

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Học phần CT205 Nhóm 01
QUẢN TRỊ CƠ SỞ DỮ LIỆU

TỔNG HỢP BÀI TẬP CUỐI KỲ

Giảng viên hướng dẫn: **PGS.TS. Nguyễn Thái Nghe**

Sinh viên thực hiện : **Lê Thái Giao Tiên**

Mã số sinh viên : **B1906586**

Học kỳ 1, 2022 - 2023

MỤC LỤC

I. TÓM TẮT LÝ THUYẾT.....	1
1.1. HQTCSDDL là gì? Các HQTCSDDL hiện nay	1
1.2. Khác nhau giữa HQTCSDDL và bảng tính (vd, Excel) là gì?	1
1.3. DBA là ai? Nhiệm vụ?	2
1.4. Các phương pháp bảo vệ dữ liệu	3
1.5. Giao dịch là gì? Tại sao cần phải giao dịch?.....	3
1.6. Các trạng thái của giao dịch	4
1.7. Thuộc tính ACID là gì?	4
1.8. Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục?	5
1.9. Lịch trình là gì? Lịch trình tuần tự và không tuần tự.....	5
1.10. Các kỹ thuật quản lý cạnh tranh.....	5
1.10.1. Bi quan (lock)	5
1.10.2. Lạc quan: 3 kỳ (đọc, kiểm tra, ghi)	6
1.10.3. Độ mịn của mục dữ liệu	7
1.11. Phục hồi dữ liệu.....	7
1.11.1. Tại sao cần phục hồi dữ liệu	7
1.11.2. Điểm kiểm tra là gì (check point).....	7
1.11.3. Các kỹ thuật phục hồi dữ liệu.....	7
II. BÀI TẬP 1 – CSDL XE Ô TÔ.....	9
2.1. Câu 1: Tạo các bảng để lưu thông tin	9
2.2. Câu 2: Viết thủ tục cho phép nhập vào 1 xe mới	22
2.3. Câu 3: Gọi thủ tục để Nhập liệu như bảng trên.....	23
2.4. Câu 4: Viết thủ tục cho phép cập nhật giá niêm yết và giá đàm phán của 1 xe nào đó. Tất cả hàm, thủ tục đều phải có lệnh test thử.	23
2.5. Câu 5: Viết hàm trả về giá đàm phán của 1 xe nào đó	24
2.6. Câu 6: Viết hàm trả về loại xe, hãng xe có giá cao nhất.....	26
2.7. Câu 7: Viết thủ tục để xóa 1 xe nào đó	28
2.8. Câu 8: Viết hàm trả về giá lăn bánh ở Cần Thơ của 1 xe nào đó, biết rằng giá lăn bánh được tính theo công thức sau.....	28
2.9. Câu 9: Viết trigger để theo dõi việc sửa giá trong bảng xe	30

2.10.	Câu 10: Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe	31
2.11.	Câu 11: Xóa tất cả các người dùng Test1, Test2, Test3,.. mà bạn đã tạo (Lưu ý KHÔNG XÓA NHẦM của người khác).....	32
2.12.	Câu 12: Tạo người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng.....	32
III.	BÀI TẬP 2 – PHARMACY DATABASE	34
3.1.	Mô tả và thiết kế CSDL về một lĩnh vực quản lý hiệu thuốc.....	34
3.1.1.	Tạo bảng	34
3.1.2.	Nhập dữ liệu cho bảng	37
3.2.	Thủ tục.....	40
3.2.1.	Thủ tục thêm 1 dược phẩm mới.....	40
3.2.2.	Thủ tục cập nhật giá của 1 dược phẩm	41
3.2.3.	Thủ tục xóa 1 dược phẩm	42
3.3.	Hàm	42
3.3.1.	Hàm trả về đơn giá của 1 dược phẩm	42
3.3.2.	Hàm trả về ngày bán của 1 đơn thuốc nào đó	43
3.3.3.	Hàm trả về thành tiền của 1 đơn thuốc nào đó	44
3.4.	Trigger	45
3.4.1.	Trigger theo dõi thêm dược phẩm mới	45
3.4.2.	Trigger theo dõi sửa giá dược phẩm.....	46
3.4.3.	Trigger theo dõi xóa dược phẩm.....	47

DANH MỤC VIẾT TẮT

Từ viết tắt	Ý nghĩa
HQTCSDL	Hệ quản trị cơ sở dữ liệu
CSDL	Cơ sở dữ liệu
DBA	Database Administrator (nhà quản trị CSDL)
GD	Giao dịch

I. TÓM TẮT LÝ THUYẾT

1.1. HQTCSDL là gì? Các HQTCSDL hiện nay

- HQTCSDL là hệ thống phần mềm cho phép người dùng định nghĩa, tạo và duy trì CSDL, đồng thời quản lý việc cung cấp dịch vụ truy cập đến CSDL này.

(Chương 1 – Trang 2)

- HQTCSDL phổ biến hiện nay: Oracle, DB2, MS SQL Server, MS Access và các phần mềm nguồn mở MySQL, MongoDB, PostGreSQL,...

(Chương 1 – Trang 2)

1.2. Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì?

- **HQTCSDL:** Là một tập hợp các chương trình để truy cập dữ liệu cung cấp thông tin về một tổ chức cụ thể. Thực hiện các hoạt động như: tạo, chèn, xóa, cập nhật và truy xuất thông tin. Nó là một hệ thống phần mềm mục đích chung tạo điều kiện thuận lợi cho quá trình xác định, xây dựng và thao tác cơ sở dữ liệu cho các ứng dụng khác nhau và liên quan đến dữ liệu.
- **Bảng tính:** Là một tệp có thể giúp tính toán và sắp xếp dữ liệu. Bảng tính có thể có một giá trị số, văn bản, công thức và hàm. Nó có các cột và hàng để giữ cho thông tin được chèn dễ đọc và dễ hiểu và là một bảng đồ thị điện tử.

HQTCSDL	Bảng tính
Có tính bảo mật với việc phân quyền quản trị người dùng.	Tính bảo mật chưa cao, không phân quyền được người dùng.
Lưu trữ lượng dữ liệu lớn.	Khó lưu trữ và truy xuất lượng dữ liệu lớn (ví dụ: truy xuất dữ liệu qua nhiều năm).
Duy trì CSDL an toàn với ràng buộc toàn vẹn.	Không có ràng buộc toàn vẹn dữ liệu.
Một tập hợp dữ liệu liên quan đến nhau	Một bảng biểu đồ điện tử với các hàng và cột
Lưu trữ và truy xuất dữ liệu	Lưu trữ và phân tích dữ liệu
Hỗ trợ cho việc tổ chức và lựa chọn dữ liệu	Hỗ trợ cho các tính toán phức tạp
Công suất chỉ bị giới hạn bởi máy tính	Chứa lượng thông tin hạn chế
Hướng tới việc liên kết và tổ chức dữ liệu nhiều hơn	Hướng tới tính toán nhiều hơn
Có thể thực hiện các tính toán đơn giản	Có thể thực hiện tổ chức và lựa chọn dữ liệu đơn giản
Xử lý dữ liệu hiệu quả	Xử lý dữ liệu không hiệu quả
Ví dụ: Oracle, MySQL, MS Access	Ví dụ: Excel, Lotus 1-2-3

(<https://www.geeksforgeeks.org/difference-between-dbms-and-spreadsheet/>)

1.3. DBA là ai? Nhiệm vụ?

DBA là **nhà quản trị CSDL** (*Database Administrator*)

Nhiệm vụ:

- Đảm bảo cho sự hoạt động thông suốt của các chức năng, tính hiệu quả và ứng dụng truy cập vào CSDL của tổ chức
- Khi cần phát triển một ứng dụng mới:
 - o DBA tham gia vào tất cả giai đoạn của đề án nhằm đảm bảo ứng dụng có sự truy cập chính xác, hiệu quả đến dữ liệu của tổ chức.
- Khi ứng dụng chuẩn bị đưa vào sử dụng:

- DBA đảm bảo HQTCSĐL đã được chuẩn bị sẵn sàng cho tải mới và cũng có trách nhiệm đưa CSDL mới từ môi trường thử nghiệm vào hoạt động.
- Khi ứng dụng đã được đưa vào sử dụng:
 - DBA đảm bảo tính sẵn sàng, an ninh, toàn vẹn của hệ thống (cài đặt các ràng buộc trên CSDL); giám sát hiệu suất, điều chỉnh, sao lưu và phục hồi hệ thống.
- Khi ứng dụng không còn được sử dụng:
 - DBA xác định tình trạng cuối cùng của dữ liệu sử dụng bởi ứng dụng để có thể cần cho ứng dụng nào khác, hay cần lưu trữ theo quy định nào đó,...

(Chương 1 – Trang 7)

1.4. Các phương pháp bảo vệ dữ liệu

- Phương pháp không sử dụng máy tính:
 - Bố trí thiết bị một cách an toàn
 - Quản trị nhân sự
 - Chính sách an ninh và kế hoạch đối phó với những bất ngờ
- Phương pháp có sử dụng máy tính:
 - Cấp quyền (authorization)
 - Khung nhìn (Views): Cho phép ẩn và hiển thị dữ liệu cần thiết
 - Sao lưu và phục hồi (Backup and restore)
 - Toàn vẹn dữ liệu (Integrity): Khóa chính (PRIMARY KEY), khóa ngoại (FOREIGN KEY), Điều kiện (CHECK), NOT NULL, giá trị mặc định (DEFAULT)
 - Ràng buộc về khóa hay ràng buộc thực thể: PRIMARY KEY, UNIQUE KEY
 - Ràng buộc về miền trị: CHECK CONSTRAINT, DEFAULT VALUE
 - Ràng buộc về tham chiếu: FOREIGN KEY
 - Ràng buộc khác: PROCEDURE, FUNCTION, TRIGGER
 - Mã hóa (Encryption)

(Chương 2 – Trang 9)

1.5. Giao dịch là gì? Tại sao cần phải giao dịch?

- **Giao dịch** là một hành động hay một chuỗi các hành động được thực hiện bởi 1 người dùng hoặc 1 chương trình ứng dụng, trong đó có truy cập hoặc thay đổi nội dung của một CSDL. Nó có thể là toàn bộ chương trình, một phần của chương trình, hoặc một lệnh đơn lẻ như INSERT hay UPDATE, và nó có thể bao gồm nhiều thao tác trên CSDL. *(Chương 3 – Trang 18)*

- **Cần phải giao dịch vì:** Lợi ích chính của việc sử dụng các giao dịch là tính toàn vẹn dữ liệu. Nhiều cách sử dụng cơ sở dữ liệu yêu cầu lưu trữ dữ liệu cho nhiều bảng hoặc nhiều hàng trong cùng một bảng để duy trì một tập dữ liệu nhất quán. Sử dụng giao dịch đảm bảo rằng các kết nối khác với cùng một cơ sở dữ liệu sẽ thấy tất cả các bản cập nhật hoặc không có bản cập nhật nào trong số chúng.

(https://www.justsoftwaresolutions.co.uk/database/database_tip_use_transactions.html)

1.6. Các trạng thái của giao dịch

- **Hoạt động (Active):** Trạng thái khởi đầu; giao dịch giữ trong trạng thái này trong khi nó đang thực hiện.
- **Hoàn tất một phần (Partially Committed):** Sau khi lệnh cuối cùng được thực hiện.
- **Thất bại (Failed):** Sau khi phát hiện rằng sự thực hiện không thể tiếp tục được nữa.
- **Hủy bỏ (Aborted):** Sau khi giao dịch bị cuộn lại (rolled back) và CSDL đã phục hồi lại trạng thái của nó trước khi khởi động giao dịch.
- **Hoàn tất (Committed):** Sau khi hoàn thành thành công giao dịch CSDL đạt tới trạng thái nhất quán mới.

(Chương 3 – Trang 19)

1.7. Thuộc tính ACID là gì?

Thuộc tính ACID là viết tắt 4 thuộc tính của các giao dịch đều phải có, bao gồm:

- **Tính nguyên tử (Atomicity):** hay tính ‘Tất cả hoặc không có gì’: Một giao dịch hoặc được thực hiện toàn bộ hoặc không thực hiện gì.
- **Tính nhất quán (Consistency):** Một giao dịch phải chuyển CSDL từ một trạng thái nhất quán này sang một trạng thái nhất quán khác. Tính nhất quán của CSDL yêu cầu chỉ những dữ liệu không vi phạm ràng buộc dữ liệu mới được ghi vào CSDL.
- **Tính cô lập (Isolation):** Các kết quả giữa chừng của một giao dịch chưa hoàn tất thì không để các giao dịch khác nhìn thấy.
- **Tính bền vững (Duration):** Mỗi một giao dịch hoàn tất, tất cả các cập nhật đã thực hiện trên CSDL vẫn còn đó, ngay cả khi sự cố hệ thống sau khi giao dịch đã hoàn tất.

(Chương 3 – Trang 20)

1.8. Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục?

- **Các vấn đề cạnh tranh** trong môi trường đa người dùng:
 - o *Mất dữ liệu đã cập nhật (lost update)*: Kết quả cập nhật dữ liệu thành công của một người dùng có thể bị ghi đè bởi một người dùng khác dẫn đến sai kết quả.
 - o *Sự phụ thuộc vào các giao dịch không hoàn tất (uncommitted dependency)*: Một GD cho phép các GD khác nhìn thấy các kết quả tạm thời trước khi nó hoàn tất.
 - o *Phân tích không nhất quán (inconsistent analysis)*: GD chỉ được phép đọc một phần kết quả của các GD chưa hoàn tất mà các GD này cùng lúc cũng đang cập nhật CSDL.
- **Khắc phục** bằng điều khiển cạnh tranh (Concurrency control): Quản lý cạnh tranh là quá trình quản lý các thao tác đang diễn ra đồng thời trên CSDL mà không cho phép chúng can thiệp lẫn nhau.
 - o ngăn không cho một GD khác đọc giá trị của mục dữ liệu đang được cập nhật cho đến khi việc cập nhật hoàn tất.

(Chương 3 – Trang 21)

1.9. Lịch trình là gì? Lịch trình tuần tự và không tuần tự

- **Lịch trình** một chuỗi các thao tác thực hiện bởi một tập hợp các GD cạnh tranh mà vẫn đảm bảo thứ tự của các thao tác trong từng GD đơn lẻ.
- **Lịch trình tuần tự (serial schedule)** là một lịch trình trong đó các thao tác của một GD được thực hiện liên tiếp nhau, không có bất kỳ thao tác nào của các GD khác xen vào giữa.
- **Lịch trình không tuần tự (nonserial schedule)** là một lịch trình trong đó các thao tác từ một tập hợp các giao dịch cạnh tranh đan xen lẫn nhau. Khi đó, các lịch trình này có thể đưa CSDL về tình trạng không nhất quán hay các kết quả không chính xác như phân tích trong các phần kế tiếp.

(Chương 3 – Trang 23)

1.10. Các kỹ thuật quản lý cạnh tranh

1.10.1. Bi quan (lock)

Cho phép các GD thực hiện song song một cách an toàn dựa trên một số ràng buộc nào đó.

- 2 phương pháp: Khóa chốt và nhãn thời gian làm cho các GD bị ngưng trệ khi chúng xung đột với các giao dịch khác tại một thời điểm nào đó trong tương lai.

- **Khóa chốt (locking):** là một thủ tục sử dụng để điều khiển truy cập cạnh tranh đến dữ liệu. Khi một GD đang truy cập vào CSDL, một khóa chốt có thể từ chối truy cập đến từ các CD khác để tránh kết quả không đúng.
 - o **Khóa đọc (read lock):** Nếu một GD có một khóa đọc trên một mục dữ liệu, nó có thể đọc nhưng không thể cập nhật mục dữ liệu đó.
 - o **Khóa ghi (write lock):** Nếu một GD có một khóa ghi trên một mục dữ liệu, nó có thể đọc và cập nhật mục dữ liệu đó.
- Bởi vì các thao tác đọc không thể xung đột, nên có thể có nhiều GD giữ khóa đọc một cách đồng thời trên cùng một mục dữ liệu. Ngược lại, một khóa ghi cho phép một GD độc quyền truy cập trên mục dữ liệu đó. Vì vậy, ngay khi một GD giữ một khóa ghi trên mục dữ liệu, thì không một GD nào khác có thể đọc hoặc cập nhật mục dữ liệu đó.

(Chương 3 – Trang 27)

1.10.2. Lạc quan: 3 kỳ (đọc, kiểm tra, ghi)

Các **kỹ thuật lạc quan (optimistic techniques)** dựa trên giả thiết rằng xung đột thì hiếm, và sẽ hiệu quả hơn nếu cho phép các GD được tiến hành mà không cần bắt chúng phải trì hoãn để đảm bảo tính khả tuần tự (Kung & Robinson, 1981).

Khi một giao dịch muốn hoàn tất, hệ thống mới kiểm tra xem liệu có xung đột đã xảy ra hay không. Nếu có xung đột, thì GD phải bị hủy và khởi động lại. Vì giả thiết là xung đột hiếm xảy ra, nên việc cuộn lại cũng sẽ hiếm.

Giao thức điều khiển cạnh tranh lạc quan: Một GD sẽ trải qua ba kỳ nếu là GD cập nhật, và trải qua 2 kỳ nếu là GD chỉ đọc:

- **Kỳ đọc:** kỳ này kéo dài từ lúc bắt đầu GD cho đến ngay trước hành động commit. GD đọc các giá trị của tất cả các mục dữ liệu nó cần từ CSDL và lưu chúng vào các biến cục bộ. Các cập nhật sẽ được áp trên bản sao chép cục bộ của dữ liệu, không phải trên CSDL.
- **Kỳ kiểm tra:** Kỳ này theo sau kỳ đọc. Các kiểm tra sẽ được thực hiện để đảm bảo tính khả tuần tự không bị vi phạm nếu các cập nhật của GD được đưa vào CSDL.

Đối với các GD chỉ đọc, điều này bao gồm việc kiểm tra xem các giá trị dữ liệu đã đọc vào các biến vẫn còn là các giá trị hiện hành trong các mục dữ liệu tương ứng. Nếu không có sự can thiệp nào đã diễn ra, thì GD được hoàn tất. Nếu không, GD phải bị hủy và khởi động lại.

Đối với một GD có cập nhật, việc kiểm tra bao gồm việc xác định liệu GD đó có đưa CSDL về một tình trạng nhất quán với tính khả tuần tự được duy trì hay không. Nếu không thì GD phải bị hủy.

- **Kỳ ghi:** Kỳ này theo sau kỳ kiểm tra thành công đối với các giao dịch cập nhật. Trong kỳ này, các cập nhật đã thực hiện trên biến cục bộ sẽ được chép vào CSDL.

(Chương 3 – Trang 35)

1.10.3. Độ mịn của mục dữ liệu

Độ mịn là kích cỡ của mục dữ liệu được chọn như là một đơn vị bảo vệ bởi giao thức điều khiển cạnh tranh. Một mục dữ liệu điển hình được chọn như là một trong những thứ có kích cỡ từ lớn đến nhỏ sau đây:

- Toàn bộ CSDL
- Một tập tin (file)
- Một trang tin (page), đôi khi còn gọi là một vùng hay một không gian CSDL – một đoạn đĩa vật lý nơi lưu trữ các quan hệ
- Một mẫu tin (record)
- Một giá trị của một trường trong một mẫu tin.

(Chương 3 – Trang 37)

1.11. Phục hồi dữ liệu

1.11.1. Tại sao cần phục hồi dữ liệu

Vì có rất nhiều nguyên nhân gây lỗi như hệ thống bị ngưng lại, lỗi thiết bị lưu trữ, lỗi phần mềm ứng dụng, thiên tai, bất cẩn hay phá hoại dữ liệu có chủ ý nên cần giảm thiểu các ảnh hưởng và phục hồi CSDL về trạng thái đúng sau sự cố có lỗi xảy ra.

(Chương 3 – Trang 39)

1.11.2. Điểm kiểm tra là gì (check point)

Điểm kiểm tra (checkpoint) là một điểm mà tại đó sự đồng bộ giữa CSDL và tập tin nhật ký GD được ghi nhận. Khi đó, tất cả các vùng đệm phải được ghi-ép-buộc ra bộ lưu trữ thứ cấp.

(Chương 3 – Trang 42)

1.11.3. Các kỹ thuật phục hồi dữ liệu

- *Nếu CSDL bị hư*, không đọc được nữa, thì ta cần sử dụng bản sao lưu cuối cùng và chạy lại các thao tác của các GD hoàn tất đã ghi nhận trong tập tin nhật ký.
- *Nếu CSDL không bị hư vật lý* nhưng trở nên không nhất quán, chẳng hạn như khi hệ thống bị treo khi đang thực hiện giao dịch, thì ta cần undo các

thay đổi gây ra tình trạng không nhất quán. Cũng có thể phải redo một số giao dịch để đảm bảo các cập nhật của nó được lưu ra bộ lưu trữ thứ cấp.

- Ba kỹ thuật để phục hồi cho trường hợp này là cập nhật trì hoãn (deferred update), cập nhật tức thì (immediate update) và tạo trang bóng (shadow paging).
 - **Cập nhật trì hoãn:** Ý tưởng chủ yếu là trì hoãn các cập nhật thực sự lên CSDL cho đến khi GD kết thúc thành công và đạt đến điểm hoàn tất.
 - **Cập nhật tức thì:** CSDL có thể được cập nhật ngay bởi các thao tác của một GD, trước khi GD tiến đến điểm hoàn tất.
 - **Tạo trang bóng:** Cơ chế này duy trì các bảng hai-trang trong suốt quá trình thực hiện của GD: một bảng trang hiện tại và một bảng trang bóng. Khi một GD bắt đầu, các bảng này giống nhau. Bảng trang bóng không bao giờ thay đổi, vì vậy được dùng để phục hồi CSDL khi có sự cố xảy ra. Trong suốt GD, bảng trang hiện tại được dùng để ghi nhận mọi thay đổi trên dữ liệu. Khi GD hoàn tất, bảng trang hiện tại trở thành bảng trang bóng.

(Chương 3 – Trang 42)

II. BÀI TẬP 1 – CSDL XE Ô TÔ

-- Le Thai Giao Tien B1906586 --

```
drop table HangXe CASCADE CONSTRAINTS;
drop table DongCo CASCADE CONSTRAINTS;
drop table DongXe CASCADE CONSTRAINTS;
drop table PhienBan CASCADE CONSTRAINTS;
drop table PhanKhucXe CASCADE CONSTRAINTS;
drop table DamPhan CASCADE CONSTRAINTS;
drop table Xe CASCADE CONSTRAINTS;
drop table ThongSoKyThuat CASCADE CONSTRAINTS;
drop table NoiLanBanh CASCADE CONSTRAINTS;
drop table GiaLanBanh CASCADE CONSTRAINTS;
```

2.1. Câu 1: Tạo các bảng để lưu thông tin

-- Hang Xe

```
create table HangXe(
    HX_Ma char(7) NOT NULL check (regexp_like (HX_Ma, 'HX\d{5}')),
    HX_Ten varchar(100) NOT NULL,
    PRIMARY KEY (HX_Ma)
);
```

-- Dong Co

```
create table DongCo(
    DC_Ma char(7) NOT NULL check (regexp_like (DC_Ma, 'DC\d{5}')),
    HX_Ma char(7),
    DC_Ten varchar(200) NOT NULL,
    PRIMARY KEY (DC_Ma),
```

```

        CONSTRAINT FK_DC_HX FOREIGN KEY (HX_Ma) REFERENCES
HangXe(HX_Ma)

```

```
);
```

```
-- Dong Xe
```

```
create table DongXe(
```

```
    DX_Ma char(7) NOT NULL check (regexp_like (DX_Ma, 'DX\d{5}')),
```

```
    HX_Ma char(7),
```

```
    DX_Ten varchar(200) NOT NULL,
```

```
    PRIMARY KEY (DX_Ma),
```

```
        CONSTRAINT FK_DX_HX FOREIGN KEY (HX_Ma) REFERENCES
HangXe(HX_Ma)

```

```
);
```

```
-- Phien Ban
```

```
create table PhienBan(
```

```
    PB_Ma char(7) NOT NULL check (regexp_like (PB_Ma, 'PB\d{5}')),
```

```
    DX_Ma char(7),
```

```
    PB_Ten varchar(100) NOT NULL,
```

```
    PRIMARY KEY (PB_Ma),
```

```
        CONSTRAINT FK_PB_HX FOREIGN KEY (DX_Ma) REFERENCES
DongXe(DX_Ma)

```

```
);
```

```
-- Phan Khuc Xe
```

```
create table PhanKhucXe(
```

```
    PKX_Ma char(7) NOT NULL check (regexp_like (PKX_Ma, 'PKX\d{4}')),
```

```
    PKX_Ten varchar(100) NOT NULL,
```

```
    PRIMARY KEY (PKX_Ma)

```

```
);
```

-- Dam Phan

```
create table DamPhan(
    DP_Ma char(7) NOT NULL check (regexp_like (DP_Ma, 'DP\d{5}')),
    DP_Ten varchar(300),
    PRIMARY KEY (DP_Ma)
);
```

```
ALTER TABLE DamPhan
    ADD DP_GiaTri float(15);
```

```
ALTER TABLE DamPhan
    MODIFY DP_GiaTri float(15) DEFAULT 0;
```

```
Select * from DamPhan;
```

-- Xe

```
create table Xe(
    X_Ma char(7) NOT NULL check (regexp_like (X_Ma, 'X\d{6}')),
    PB_Ma char(7),
    PKX_Ma char(7),
    DC_Ma char(7),
    DP_Ma char(7),
    X_GiaNiemYet float(20) NOT NULL CHECK (X_GiaNiemYet > 0),
    PRIMARY KEY (X_Ma),
    CONSTRAINT FK1_X_PB FOREIGN KEY (PB_Ma) REFERENCES
PhienBan(PB_Ma),
    CONSTRAINT FK2_X_PKX FOREIGN KEY (PKX_Ma) REFERENCES
PhanKhucXe(PKX_Ma),
    CONSTRAINT FK3_X_DC FOREIGN KEY (DC_Ma) REFERENCES
DongCo(DC_Ma),
```

```

        CONSTRAINT FK4_X_DP FOREIGN KEY (DP_Ma) REFERENCES
DamPhan(DP_Ma)
);

ALTER TABLE Xe
    MODIFY DP_Ma char(7) DEFAULT 'DP00000';
commit;

-- Thông Số Kỹ Thuật
create table ThôngSốKỹThuật(
    TSKT_Ma char(7) NOT NULL check (regexp_like (TSKT_Ma, 'TSKT\d{3}')),
    X_Ma char(7),
    TSKT_CongSuat varchar(50),
    TSKT_MomenXoan varchar(200),
    TSKT_HopSo varchar(50),
    TSKT_HeDanDong varchar(50),
    TSKT_SoCho int NOT NULL CHECK (TSKT_SoCho > 0),
    PRIMARY KEY (TSKT_Ma),
    CONSTRAINT FK_TSKT_X FOREIGN KEY (X_Ma) REFERENCES
Xe(X_Ma)
);--ALTER TABLE ThôngSốKỹThuật MODIFY TSKT_MomenXoan varchar(200);

-- Nội Lan Bánh
create table NộiLanBánh(
    NLB_Ma char(7) NOT NULL check (regexp_like (NLB_Ma, 'NLB\d{4}')),
    NLB_Ten varchar(100) NOT NULL,
    PRIMARY KEY (NLB_Ma)
);

ALTER TABLE NộiLanBánh
    ADD NLB_GiaTri float(15); -- chi số % nhận với giá niêm yết

```



```
select * from NoiLanBanh;
```

```
-- Gia Lan Banh
```

```
create table GiaLanBanh(
```

```
    GLB_Ma char(7) NOT NULL check (regexp_like (GLB_Ma, 'GLB\d{4}')),
```

```
    X_Ma char(7),
```

```
    NLB_Ma char(7),
```

```
    GLB_PhiTruocBa float(20) DEFAULT NULL,
```

```
    GLB_PhiSDDuongBo float(20) DEFAULT 1560000,
```

```
    GLB_BaoHiem float(20) DEFAULT 437000,
```

```
    GLB_PhiDKBienSo float(20) DEFAULT 20000000,
```

```
    GLB_PhiDangKiem float(20) DEFAULT 340000,
```

```
    GLB_TongCong float(20) DEFAULT NULL,
```

```
    CONSTRAINT PK_GLB PRIMARY KEY(GLB_Ma, X_Ma, NLB_Ma),
```

```
    CONSTRAINT FK1_GLB_X FOREIGN KEY (X_Ma) REFERENCES  
Xe(X_Ma),
```

```
    CONSTRAINT FK2_GLB_NLB FOREIGN KEY (NLB_Ma) REFERENCES  
NoiLanBanh(NLB_Ma)
```

```
);
```

```
-----  
-----  
-----  
-----
```

```
-- Hang Xe
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00001', 'Aston Martin');
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00002', 'Audi');
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00003', 'Bentley');
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00004', 'BMW');
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00005', 'Ford');
```

```
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00006', 'Honda');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00007', 'Hyundai');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00008', 'Isuzu');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00009', 'Jaguar');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00010', 'Jeep');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00011', 'Kia');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00012', 'Land Rover');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00013', 'Lexus');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00014', 'Maserati');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00015', 'Mazda');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00016', 'Mercedes');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00017', 'MG');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00018', 'Mini');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00019', 'Mitsubishi');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00020', 'Nissan');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00021', 'Peugeot');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00022', 'Porsche');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00023', 'Ram');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00024', 'Subaru');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00025', 'Suzuki');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00026', 'Toyota');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00027', 'Vinfast');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00028', 'Volkswagen');
INSERT INTO HangXe (HX_Ma, HX_Ten) VALUES ('HX00029', 'Volvo');
commit;
select * from HangXe;
```

-- Dong Co

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00001',
'HX00002', 'TFSI 2.0 Mild-hybrid 12V');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00002',
'HX00002', 'TFSI 3.0 Mild-hybrid 48V');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00003',
'HX00003', '4.0 Twin-turbocharged V8 TSI');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00004',
'HX00003', '4.0 V8');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00005',
'HX00018', 'I4, TwinPower Turbo');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00006',
'HX00018', 'Xang, I4, TwinPower Turbo');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00007',
'HX00022', 'Boxer H-4 Tang ap');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00008',
'HX00022', 'Boxer H-6, twin-turbo');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00009',
'HX00027', 'Xang 1.4, 4 xi lanh thang hang');

INSERT INTO DongCo (DC_Ma, HX_Ma, DC_Ten) VALUES ('DC00010',
'HX00027', 'I-4, DOHC, tang ap, van bien thien');

commit;

select * from DongCo;

-- Dong Xe

INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00001',
'HX00002', 'A4 2021');

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00002',
'HX00002', 'A6 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00003',
'HX00002', 'A7 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00004',
'HX00003', 'Flying Spur 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00005',
'HX00003', 'Continental 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00006',
'HX00018', '3 Door 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00007',
'HX00018', '5 Door 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00008',
'HX00022', '718 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00009',
'HX00022', '911 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00010',
'HX00027', 'Fadil 2021');
```

```
INSERT INTO DongXe (DX_Ma, HX_Ma, DX_Ten) VALUES ('DX00011',
'HX00027', 'Lux A2.0 2021');
```

```
commit;
```

```
select * from DongXe;
```

```
-----
-----
```

```
-- Phien Ban
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00001',
'DX00001', '45 TFSI quattro');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00002',
'DX00001', '40 TFSI Advanced Plus');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00003',
'DX00001', '40 TFSI Advanced');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00004',  
'DX00002', '45 TFSI');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00005',  
'DX00003', '55 TFSI quattro');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00006',  
'DX00004', 'V8');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00007',  
'DX00005', 'GT V8');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00008',  
'DX00006', 'Cooper S 3 c?a');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00009',  
'DX00007', 'Cooper S 5 c?a');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00010',  
'DX00008', 'Cayman');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00011',  
'DX00008', 'Boxster');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00012',  
'DX00008', 'Cayman T');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00013',  
'DX00008', 'Boxster T');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00014',  
'DX00008', 'Cayman S');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00015',  
'DX00008', 'Boxster S');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00016',  
'DX00009', 'Carerra');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00017',  
'DX00009', 'Carerra 4');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00018',  
'DX00009', 'Carerra Cabriolet');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00019',  
'DX00009', 'Carerra S');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00020',
'DX00009', 'Carerra 4 Cabriolet');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00021',
'DX00009', 'Targa 4');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00022',
'DX00010', 'Tieu chuan');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00023',
'DX00010', 'Nang cao');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00024',
'DX00010', 'Cao cap');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00025',
'DX00011', 'Tieu chuan');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00026',
'DX00011', 'Nang cao');
```

```
INSERT INTO PhienBan (PB_Ma, DX_Ma, PB_Ten) VALUES ('PB00027',
'DX00011', 'Cao cap');
```

```
commit;
```

```
select * from PhienBan;
```

```
-----
-----
```

```
-- Phan Khuc Xe
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0001', 'Xe sang
co nho');
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0002', 'Xe sang
co trung');
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0003', 'Sieu
sang co lon');
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0004', 'Sieu xe
/ Xe the thao');
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0005', 'Xe nho
co A');
```

```
INSERT INTO PhanKhucXe (PKX_Ma, PKX_Ten) VALUES ('PKX0006', 'Xe co
trung hang E');
```

```
commit;
```

```
select * from PhanKhucXe;
```

```
-----
-----
```

```
-- Dam Phan
```

```
INSERT INTO DamPhan (DP_Ma, DP_Ten) VALUES ('DP00001', 'Khuyen mai');
```

```
INSERT INTO DamPhan (DP_Ma, DP_Ten) VALUES ('DP00002', 'Tang 10 - 40
trieu dong');
```

```
INSERT INTO DamPhan (DP_Ma, DP_Ten) VALUES ('DP00003', 'Tang 10 - 30
trieu dong');
```

```
INSERT INTO DamPhan (DP_Ma, DP_Ten) VALUES ('DP00004', 'Tang 10% khi
mua xe tra thang hoac ho tro lai xuất ngân hàng khi tra gop');
```

```
INSERT INTO DamPhan (DP_Ma, DP_Ten, DP_GiaTri) VALUES ('DP00000',
'Khong dam phan', DEFAULT);
```

```
UPDATE DamPhan
```

```
    SET DP_GiaTri = DEFAULT
```

```
    WHERE DP_Ma = 'DP00001';
```

```
UPDATE DamPhan
```

```
    SET DP_GiaTri = DEFAULT
```

```
    WHERE DP_Ma = 'DP00003';
```

```
UPDATE DamPhan
```

```
    SET DP_GiaTri = DEFAULT
```

```
    WHERE DP_Ma = 'DP00004';
```

```
commit;
```

```
select * from DamPhan;
```


-- Xe

INSERT INTO Xe (X_Ma, PB_Ma, PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)
VALUES ('X000001', 'PB00006', 'PKX0003', 'DC00003', NULL, 19500000000);

INSERT INTO Xe (X_Ma, PB_Ma, PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)
VALUES ('X000002', 'PB00008', 'PKX0001', 'DC00005', 'DP00002', 18390000000);

INSERT INTO Xe (X_Ma, PB_Ma, PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)
VALUES ('X000003', 'PB00010', 'PKX0004', 'DC00007', NULL, 36000000000);

INSERT INTO Xe (X_Ma, PB_Ma, PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)
VALUES ('X000004', 'PB00022', 'PKX0005', 'DC00009', 'DP00004', 4250000000);

UPDATE Xe

SET DP_Ma = DEFAULT

WHERE X_Ma = 'X000001';

UPDATE Xe

SET DP_Ma = DEFAULT

WHERE X_Ma = 'X000003';

commit;

select * from Xe;

-- Thong So Ky Thuat

INSERT INTO ThongSoKyThuat VALUES ('TSKT001', 'X000001', '542/6000',
'770/2000-4500', '8 cap ly hop kep', 'AWD', 5);

INSERT INTO ThongSoKyThuat VALUES ('TSKT002', 'X000002', '192 hp/5000 -
6000', '280 Nm/1350 - 4600', 'Ly hop kep 7 cap Steptronic', '4 banh', 4);

INSERT INTO ThongSoKyThuat VALUES ('TSKT003', 'X000003', '300/6500,
350/6500', '380/2050 - 4500, 380/2150 - 4500, 420/1900-4500, 420/1900 - 4500',
'PDK 7 cap, 6MT', 'Banh sau', 2);


```
INSERT INTO ThôngSốKỹThuật VALUES ('TSKT004', 'X000004', '98/6200',
'128/4400', 'Tu đing vo cap - CVT', 'Cau truoc - FWD', 5);
```

```
commit;
```

```
delete from ThôngSốKỹThuật;
```

```
select * from ThôngSốKỹThuật;
```

```
-----
-----
```

```
-- Noi Lan Banh
```

```
INSERT INTO NoiLanBanh (NLB_Ma, NLB_Ten) VALUES ('NLB0001', 'Ha Noi');
```

```
INSERT INTO NoiLanBanh (NLB_Ma, NLB_Ten) VALUES ('NLB0002', 'TP
HCM');
```

```
INSERT INTO NoiLanBanh (NLB_Ma, NLB_Ten) VALUES ('NLB0003', 'Ha Tinh');
```

```
INSERT INTO NoiLanBanh (NLB_Ma, NLB_Ten) VALUES ('NLB0004', 'Cac tinh
thanh khac');
```

```
UPDATE NoiLanBanh
```

```
    SET NLB_GiaTri = 0.12
```

```
    WHERE NLB_Ma = 'NLB0001';
```

```
UPDATE NoiLanBanh
```

```
    SET NLB_GiaTri = 0.1
```

```
    WHERE NLB_Ma = 'NLB0002';
```

```
UPDATE NoiLanBanh
```

```
    SET NLB_GiaTri = 0.11
```

```
    WHERE NLB_Ma = 'NLB0003';
```

```
UPDATE NoiLanBanh
```

```
    SET NLB_GiaTri = 0.1
```

```
    WHERE NLB_Ma = 'NLB0004';
```

```
select * from noilanbanh;
```

```
commit;
```


 -- Gia Lan Banh

INSERT INTO GiaLanBanh VALUES ('GLB0001', 'X000001', 'NLB0001', NULL,
 DEFAULT, DEFAULT, DEFAULT, DEFAULT, NULL);

INSERT INTO GiaLanBanh VALUES ('GLB0002', 'X000002', 'NLB0001', NULL,
 DEFAULT, DEFAULT, DEFAULT, DEFAULT, NULL);

commit;

select * from GiaLanBanh;

2.2. Câu 2: Viết thủ tục cho phép nhập vào 1 xe mới

-- Cau 2 Viet thu tuc cho phép nhap mot xe moi -----

CREATE OR REPLACE PROCEDURE Insert_Xe(

 p_x_ma IN Xe.x_ma%TYPE,

 p_pb_ma IN Xe.pb_ma%TYPE,

 p_pqx_ma IN Xe.pqx_ma%TYPE,

 p_dc_ma IN Xe.dc_ma%TYPE,

 p_dp_ma IN Xe.dp_ma%TYPE,

 p_x_gianiemyet IN Xe.x_gianiemyet%TYPE)

IS

BEGIN

 INSERT INTO Xe (X_Ma, PB_Ma, PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)

 VALUES (p_x_ma, p_pb_ma, p_pqx_ma, p_dc_ma, p_dp_ma,
 p_x_gianiemyet);

 COMMIT;

END;

2.3. Câu 3: Gọi thủ tục để Nhập liệu như bảng trên

-- Cau 3 Goi thu tuc nhap

select * from Xe ORDER BY Xe.X_Ma;

EXECUTE Insert_Xe ('X000005', 'PB00007', 'PKX0003', 'DC00004', 'DP00000',
180000000000);

commit;

2.4. Câu 4: Viết thủ tục cho phép cập nhật giá niêm yết và giá đàm phán của 1 xe nào đó. Tất cả hàm, thủ tục đều phải có lệnh test thử.

-- Cau 4 Viet thu tuc cho phep cap nhat gia niem yet va gia dam phan cua mo xe nao do. Tat ca ham, thu tuc deu phai co lenh test thu.

--- ??? Viet duoc thu tuc cap nhat gia niem yet. Khong viet duoc thu tuc cap nhat gia dam phan

--- Thu tuc cap nhat gia niem yet -----

CREATE OR REPLACE PROCEDURE Update_GiaNiemYet (

 p_x_ma IN Xe.x_ma%TYPE,

 p_x_gianiemyet IN Xe.x_gianiemyet%TYPE

)

IS

BEGIN

 UPDATE Xe SET X_GiaNiemYet = p_x_gianiemyet where X_Ma = p_x_ma;

 COMMIT;

END;

--- Thuc thi thu tuc -----

```
EXECUTE Update_GiaNiemYet('X000005', 180000000000);
select * from Xe;
```

```
--- Thu tuc cap nhat gia dam phan -----
-----
```

```
/*CREATE OR REPLACE PROCEDURE Update_GiaDamPhan(
    p_x_ma IN Xe.x_ma%TYPE,
    p_dp_giatri IN DamPhan.dp_giatri%TYPE
)
IS
BEGIN
    UPDATE DamPhan SET DP_GiaTri = p_dp_giatri WHERE p_x_ma in(select
x_ma from Xe x join DamPhan d on x.DP_Ma = d.DP_Ma);
    COMMIT;
END;
```

```
select * from DamPhan;
select * from Xe;
EXECUTE Update_GiaDamPhan('X000002', 10000001);
*/
```

```
commit;
```

2.5. Câu 5: Viết hàm trả về giá đàm phán của 1 xe nào đó

```
-- Cau 5 Viet ham tra ve gia dam phan cua mot xe nao do
-- Select tra ve noi dung dam phan cua 1 xe
select X_Ma, DP_Ten
    from Xe x join DamPhan d on x.DP_Ma = d.DP_Ma
    where x_Ma = 'X000002';
```

```
-- Ham tra ve ten dam phan cua 1 xe
CREATE OR REPLACE FUNCTION Get_TenDamPhan(f_x_ma char)
RETURN varchar
IS
    temp varchar(300);
BEGIN
    select DP_Ten INTO temp
        from Xe x join DamPhan d on x.DP_Ma = d.DP_Ma
        where x_Ma = f_x_ma;
    RETURN temp;
END;

--
Select Get_TenDamPhan ('X000002') from dual;
-----

-- Cap nhat gia tri dam phan
UPDATE DamPhan
    SET DP_GiaTri = 10000000
    WHERE DP_Ma = 'DP00002';

-- Select gia dam phan cua 1 xe
Select X_Ma, DP_Ten, (X_GiaNiemYet - DP_GiaTri) AS GiaDamPhan
    from Xe x join DamPhan d on x.DP_Ma = d.DP_Ma
    where X_Ma = 'X000002';-- Gia niem yet - Gia tri dam phan

-- Select gia niem yet cua 1 xe
select X_Ma, X_GiaNiemYet from Xe where X_Ma = 'X000002';
```

-- Ham tra ve gia dam phan cua 1 xe -----

```
CREATE OR REPLACE FUNCTION Get_GiaDamPhan(f_x_ma char)
RETURN varchar
IS
    temp float(15);
BEGIN
    select (X_GiaNiemYet - DP_GiaTri) INTO temp
    from Xe x join DamPhan d on x.DP_Ma = d.DP_Ma
    where x_Ma = f_x_ma;
    RETURN temp;
END;
```

drop function Get_GiaDamPhan;

-- Thuc thi ham -----

```
Select Get_GiaDamPhan ('X000002') as GiaDamPhan from dual;
Select Get_GiaDamPhan ('X000003') as GiaDamPhan from dual;
select * from xe;
```

2.6. Câu 6: Viết hàm trả về loại xe, hãng xe có giá cao nhất

-- Cau 6 Viet ham tra ve loai xe, hang xe co gia cao nhat

--- ??? select nhung khong viet duoc ham

-- select gia niem yet cao nhat

```
select Max(X_GiaNiemYet) from Xe;
```

-- select ma xe co gia cao nhat

```
select X_Ma from Xe where X_GiaNiemYet in (select Max(X_GiaNiemYet) from
Xe);
```

```
select * from Xe;
```

```
-- select hang xe, dong xe, phien ban cua xe co gia niem yet cao nhat
```

```
select HX_Ten, DX_Ten, PB_Ten, X_Ma, X_GiaNiemYet
  from HangXe h join DongXe d on h.HX_Ma = d.HX_Ma
    join PhienBan p on d.DX_Ma = p.DX_Ma
    join Xe x on p.PB_Ma = x.PB_Ma
  where X_GiaNiemYet in (select Max(X_GiaNiemYet) from Xe);
```

```
-- Ham tra ve loai xe, hang xe có gia cao nhat-----
-----
```

```
/*CREATE TYPE x_type as object(
```

```
  HX_Ten varchar(100),
  DX_Ten varchar(100),
  PB_Ten varchar(100),
  X_Ma char(7),
  X_GiaNiemYet float(20)
```

```
);
```

```
CREATE OR REPLACE FUNCTION Get_MaxGia(loc in number)
```

```
RETURN x_type
```

```
AS
```

```
  x_record x_type;
```

```
BEGIN
```

```
  select HX_Ten, DX_Ten, PB_Ten, X_Ma, X_GiaNiemYet INTO x_record
  from HangXe h join DongXe d on h.HX_Ma = d.HX_Ma
    join PhienBan p on d.DX_Ma = p.DX_Ma
    join Xe x on p.PB_Ma = x.PB_Ma
```

```

        where X_GiaNiemYet in (select Max(X_GiaNiemYet) from Xe);
    RETURN(x_record);
END;*/

```

2.7. Câu 7: Viết thủ tục để xóa 1 xe nào đó

-- Cau 7 Viet thu tuc de xoa mot xe nao do

-- Thu tuc xoa 1 xe -----

```

CREATE OR REPLACE PROCEDURE Delete_Xe (
    p_x_ma IN Xe.x_ma%TYPE
)
IS
BEGIN
    DELETE Xe WHERE X_Ma = p_x_ma;
    COMMIT;
END;

```

-- Thuc thi thu tuc xoa 1 xe -----

```

EXECUTE Delete_Xe ('X000005');
select * from Xe;
commit;

```

2.8. Câu 8: Viết hàm trả về giá lăn bánh ở Cần Thơ của 1 xe nào đó, biết rằng giá lăn bánh được tính theo công thức sau

-- Cau 8 Viet ham tra ve gia lan banh o Can Tho cua mot xe nao do, biet rang gia lan banh duoc tinh theo cong thuc sau

--- ??? Select ket qua dung, ham tra ve sai ket qua

-- Select gia lan banh cua mot xe


```
select X_GiaNiemYet + (X_GiaNiemYet * NLB_GiaTri) + GLB_PhiSDDuongBo +
GLB_BaoHiem + GLB_PhiDKBienSo + GLB_PhiDangKiem as GiaLanBanh
```

```
from Xe x join GiaLanBanh g on x.X_Ma = g.X_Ma
```

```
join NoiLanBanh n on n.NLB_Ma = g.NLB_Ma
```

```
where g.X_Ma = 'X000001';
```

```
select * from xe;
```

```
select * from gialanbanh;
```

```
select * from noilanbanh;
```

```
-- Ham tra ve gia lan banh cua 1 xe -----
-----
```

```
----- select dung - ham khong dung
```

```
CREATE OR REPLACE FUNCTION Get_GiaLanBanh(f_x_ma char)
```

```
RETURN varchar
```

```
IS
```

```
temp float(20);
```

```
BEGIN
```

```
select (X_GiaNiemYet + (X_GiaNiemYet * NLB_GiaTri) +
GLB_PhiSDDuongBo + GLB_BaoHiem + GLB_PhiDKBienSo +
GLB_PhiDangKiem) INTO temp
```

```
from Xe x join GiaLanBanh g on x.X_Ma = g.X_Ma
```

```
join NoiLanBanh n on n.NLB_Ma = g.NLB_Ma
```

```
where g.X_Ma = f_x_ma;
```

```
RETURN temp;
```

```
END;
```

```
--drop function Get_GiaLanBanh;
```

```
-- Thuc thi ham -----
-----
```

```
Select Get_GiaLanBanh ('X000001') as GiaLanBanh from dual;
```

2.9. Câu 9: Viết trigger để theo dõi việc sửa giá trong bảng xe

-- Cau 9 Viet trigger de theo doi viec sua gia trong bang xe

--- Tao bang luu thong tin khi du lieu duoc sua doi

```
CREATE TABLE Change_Gia(
    UserName VARCHAR2(20),
    MODIFY_TIME DATE,
    X_Ma char(7),
    Old_Gia float(20),
    New_Gia float(20)
);
```

-- Tao trigger theo doi sua gia trong bang xe -----

```
CREATE OR REPLACE TRIGGER store_change_Gia
AFTER UPDATE OF X_GiaNiemYet on Xe
FOR EACH ROW
BEGIN
```

-- User la mot ham co san trong he thong cho biet ai ket noi CSDL

```
INSERT INTO Change_Gia(UserName, MODIFY_TIME, X_Ma, Old_Gia,
New_Gia)
```

```
VALUES (User,
Sysdate,:new.X_Ma,:old.X_GiaNiemYet,:new.X_GiaNiemYet);
```

```
END;
```

```
select * from change_gia;
```

```
select * from Xe;
```

-- Cap nhat gia xe de kiem tra thu trigger

```
update Xe set X_GiaNiemYet = 426000000 where X_Ma = 'X000004';
```

update Xe set X_GiaNiemYet = 425000000 where X_Ma = 'X000004';

2.10. Câu 10: Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe

-- Cau 10 Viet trigger de theo doi viec xoa du lieu trong bang xe

--- Tao bang luu thong tin khi du lieu duoc sua doi-----

CREATE TABLE Trigger_Delete_Xe(

 UserName VARCHAR2(20),

 MODIFY_TIME DATE,

 X_Ma char(7),

 PB_Ma char(7),

 PKX_Ma char(7),

 DC_Ma char(7),

 DP_Ma char(7),

 X_GiaNiemYet float(20)

);

-- Tao trigger theo doi xoa du lieu trong bang xe -----

CREATE OR REPLACE TRIGGER store_delete_Xe

AFTER DELETE on Xe

FOR EACH ROW

BEGIN

-- User la mot ham co san trong he thong cho biet ai ket noi CSDL

 INSERT INTO Trigger_Delete_Xe(UserName, MODIFY_TIME, X_Ma, PB_Ma,
 PKX_Ma, DC_Ma, DP_Ma, X_GiaNiemYet)

 VALUES (User,
Sysdate,:old.X_Ma,:old.PB_Ma,:old.PKX_Ma,:old.DC_Ma,:old.DP_Ma,:old.X_Gia
NiemYet);

END;

commit;

select * from Xe;

-- Select bang theo doi xoa xe

select * from trigger_delete_xe;

-- Xoa 1 xe de kiem tra thu trigger

delete from Xe where X_Ma = 'X000005';

2.11. Câu 11: Xóa tất cả các người dùng Test1, Test2, Test3,.. mà bạn đã tạo (Lưu ý KHÔNG XÓA NHẦM của người khác)

-- Cau 11 Xoa tat ca cac nguoi dung Test1, Test2, Test3,... ma ban da tao (Luu y KHONG XOA NHAM cua nguoi khac)

DROP USER B1906586_Test1 CASCADE;

DROP USER B1906586_Test2 CASCADE;

DROP USER B1906586_Test3 CASCADE;

2.12. Câu 12: Tạo người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng

-- Cau 12 Tao nguoi dung moi de quan ly CSDL O to tren, cap quyen tuong ung

-- Tao nguoi dung B1906586_Test1-----

create user B1906586_Test1 identified by "B1906586@Test1";

GRANT CREATE SESSION TO B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON HangXe TO B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON DongCo TO B1906586_Test1;

```
GRANT SELECT, INSERT, UPDATE, DELETE ON DongXe TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON PhienBan TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON PhanKhucXe TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON DamPhan TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON Xe TO B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON ThongSoKyThuat TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON NoiLanBanh TO
B1906586_Test1;

GRANT SELECT, INSERT, UPDATE, DELETE ON GiaLanBanh TO
B1906586_Test1;

--revoke select, insert, update, delete on Xe from B1906586_Test1;

ROLLBACK;

commit;--

--
```

III. BÀI TẬP 2 – PHARMACY DATABASE

-- Le Thai Giao Tien B1906586 --

select sysdate from dual;

-- Mô tả và thiết kế CSDL về một lĩnh vực nào đó (bán hàng, qđ sinh viên, quản lý quán cafe, du lịch,...) gồm ít nhất 5 bảng.

-- Mọi cột trong bảng phải có các ràng buộc (Khoa chính, khoa ngoại, RB duy nhất, RB kiểm tra, not null, mặc định, biểu thức chính quy - vd Mã hàng: MH01,...)

-- Nhập dữ liệu cho bảng (ít nhất 5 dòng)

-- Tên bảng (các đối tượng khác) bắt đầu là "BT_..."

-- Viết ít nhất 3 hàm, 3 thủ tục, 3 trigger để quản lý dữ liệu trên

--

drop table BT_KhachHang CASCADE CONSTRAINTS;

drop table BT_DuocSi CASCADE CONSTRAINTS;

drop table BT_DanhMuc CASCADE CONSTRAINTS;

drop table BT_DuocPham CASCADE CONSTRAINTS;

drop table BT_DonThuoc CASCADE CONSTRAINTS;

drop table BT_ChiTietDonThuoc CASCADE CONSTRAINTS;

3.1. Mô tả và thiết kế CSDL về một lĩnh vực quản lý hiệu thuốc

-- Mô tả và thiết kế CSDL về một lĩnh vực quản lý hiệu thuốc

3.1.1. Tạo bảng

-- Khách Hàng

CREATE TABLE BT_KhachHang(

KH_Ma char(7) NOT NULL check (regexp_like (KH_Ma, 'KH\d{5}')),

KH_SoDienThoai varchar(11) check (regexp_like (KH_SoDienThoai, '0\d{9,10}')),-- Bắt đầu số 0 tiếp theo là 9 hoặc 10 số

```

        KH_Ten varchar(30) NOT NULL,
        KH_DiaChi varchar(300),
        PRIMARY KEY(KH_Ma)
    );

-- Duoc Si
CREATE TABLE BT_DuocSi(
    DS_Ma char(7) NOT NULL check (regexp_like (DS_Ma, 'DS\d{5}')),
    DS_ten varchar(30) NOT NULL,
    DS_DiaChi varchar(300),
    DS_Email varchar(100) check (regexp_like (DS_Email, '^([A-Za-z]+[A-Za-z0-9.]++@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$)'),
    DS_MatKhau varchar(30),
    PRIMARY KEY(DS_Ma)
);

-- Danh Muc
CREATE TABLE BT_DanhMuc(
    DM_Ma char(7) NOT NULL check (regexp_like (DM_Ma, 'DM\d{5}')),
    DM_Ten varchar(200) NOT NULL,
    PRIMARY KEY(DM_Ma)
);

-- Duoc Pham
CREATE TABLE BT_DuocPham(
    DP_Ma char(7) NOT NULL check (regexp_like (DP_Ma, 'DP\d{5}')),
    DM_Ma char(7),
    DP_Ten varchar(200) NOT NULL,
    DP_MoTa varchar(300),
    DP_Gia float(15) CHECK(DP_Gia > 0),

```

```

        PRIMARY KEY (DP_Ma),
        CONSTRAINT FK_DP_DM FOREIGN KEY (DM_Ma) REFERENCES
BT_DanhMuc(DM_Ma)
);
-- Don Thuoc
CREATE TABLE BT_DonThuoc(
    DT_Ma char(7) NOT NULL check (regexp_like (DT_Ma, 'DT\d{5}')),
    KH_Ma char(7),
    DS_Ma char(7),
    DT_SoLuong int CHECK(DT_SoLuong > 0),
    DT_Ngay date,
    DT_ThanhTien float(15) DEFAULT NULL,
    PRIMARY KEY (DT_Ma),
    CONSTRAINT FK1_DT_KH FOREIGN KEY (KH_Ma) REFERENCES
BT_KhachHang(KH_Ma),
    CONSTRAINT FK2_DT_DS FOREIGN KEY (DS_Ma) REFERENCES
BT_DuocSi(DS_Ma)
);
-- Chi Tiet Don Thuoc
CREATE TABLE BT_ChiTietDonThuoc(
    DT_Ma char(7),
    DP_Ma char(7),
    CTDT_SoLuong int CHECK(CTDT_SoLuong > 0),
    CTDT_Gia float(15) DEFAULT NULL,
    CONSTRAINT PK_CTDT PRIMARY KEY(DT_Ma, DP_Ma),
    CONSTRAINT FK1_CTDT_DT FOREIGN KEY (DT_Ma) REFERENCES
BT_DonThuoc(DT_Ma),
    CONSTRAINT FK2_CTDT_DP FOREIGN KEY (DP_Ma) REFERENCES
BT_DuocPham(DP_Ma)

```


);

commit;--

3.1.2. Nhập dữ liệu cho bảng

-- Nhập du lieu cho bang (it nhat 5 dong)

-- Khach Hang

INSERT INTO BT_KhachHang (KH_Ma, KH_SoDienThoai, KH_Ten, KH_DiaChi)
VALUES ('KH00001', '0101010101', 'Bui Anh Tuan', 'Ha Noi');

INSERT INTO BT_KhachHang (KH_Ma, KH_SoDienThoai, KH_Ten, KH_DiaChi)
VALUES ('KH00002', '0202020202', 'Bui Nguyen Trung Quan', 'Da Lat');

INSERT INTO BT_KhachHang (KH_Ma, KH_SoDienThoai, KH_Ten, KH_DiaChi)
VALUES ('KH00003', '0303030303', 'Truc Nhan', 'Binh Dinh');

INSERT INTO BT_KhachHang (KH_Ma, KH_SoDienThoai, KH_Ten, KH_DiaChi)
VALUES ('KH00004', '0404040404', 'Giao', 'Soc Trang');

INSERT INTO BT_KhachHang (KH_Ma, KH_SoDienThoai, KH_Ten, KH_DiaChi)
VALUES ('KH00005', '0505050505', 'Tien', 'Vinh Long');

select * from BT_KhachHang;

commit;

-- Duoc Si

INSERT INTO BT_DuocSi (DS_Ma, DS_Ten, DS_DiaChi, DS_Email,
DS_MatKhau) VALUES ('DS00001', 'Vo Hoang Due', 'Can Tho', 'due@gmail.com',
'ds00001');

INSERT INTO BT_DuocSi (DS_Ma, DS_Ten, DS_DiaChi, DS_Email,
DS_MatKhau) VALUES ('DS00002', 'Trinh Duc Toan', 'Can Tho',
'toan@gmail.com', 'ds00002');

```
INSERT INTO BT_DuocSi (DS_Ma, DS_Ten, DS_DiaChi, DS_Email,
DS_MatKhau) VALUES ('DS00003', 'Moc Minh Ngoc', 'Vinh Long',
'ngoc@gmail.com', 'ds00003');
```

```
INSERT INTO BT_DuocSi (DS_Ma, DS_Ten, DS_DiaChi, DS_Email,
DS_MatKhau) VALUES ('DS00004', 'Phan Tuy Anh', 'Soc Trang',
'anh@gmail.com', 'ds00004');
```

```
INSERT INTO BT_DuocSi (DS_Ma, DS_Ten, DS_DiaChi, DS_Email,
DS_MatKhau) VALUES ('DS00005', 'Yen Xuan Van', 'Vinh Long',
'van@gmail.com', 'ds00005');
```

```
select * from BT_DuocSi;
```

```
commit;
```

```
-----
-----
```

```
-- Danh Muc
```

```
INSERT INTO BT_DanhMuc (DM_Ma, DM_Ten) VALUES ('DM00001', 'Khang
Viem');
```

```
INSERT INTO BT_DanhMuc (DM_Ma, DM_Ten) VALUES ('DM00002', 'Da lieu');
```

```
INSERT INTO BT_DanhMuc (DM_Ma, DM_Ten) VALUES ('DM00003', 'Tieu hoa');
```

```
INSERT INTO BT_DanhMuc (DM_Ma, DM_Ten) VALUES ('DM00004', 'Ha sot');
```

```
INSERT INTO BT_DanhMuc (DM_Ma, DM_Ten) VALUES ('DM00005', 'Co xuong
khop');
```

```
select * from BT_DanhMuc;
```

```
commit;
```

```
-----
-----
```

```
-- Duoc Pham
```

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES ('DP00001', 'DM00001', 'Amcinol-Paste', 'Dung ngoai: cac chung viem
dau o mieng, loi va moi.', 10000);
```

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES ('DP00002', 'DM00001', 'Axcel Diphenhydramine Paediatric Syrup', 'Giảm ho, giảm viêm mũi dị ứng và viêm phế quản dị ứng', 26500);
```

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES ('DP00003', 'DM00002', 'Avircrem', 'Các trường hợp nhiễm virus Herpes simplex trên da', 25000);
```

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES ('DP00004', 'DM00002', 'Axcel Fungicort Cream', 'Điều trị viêm da do nấm, tổn thương do côn trùng đốt', 32000);
```

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES ('DP00005', 'DM00003', 'Agimoti Agimexpharm', 'Buồn nôn và nôn nặng, cảm giác chướng và nặng vùng thượng vị', 32000);
```

```
select * from BT_DuocPham;
```

```
commit;
```

```
-----
-----
```

```
-- Đơn thuốc
```

```
INSERT INTO BT_DonThuoc (DT_Ma, KH_Ma, DS_Ma, DT_SoLuong, DT_Ngay,
DT_ThanhTien) VALUES ('DT00001', 'KH00001', 'DS00001', 1, '2022-10-04',
NULL);
```

```
INSERT INTO BT_DonThuoc (DT_Ma, KH_Ma, DS_Ma, DT_SoLuong, DT_Ngay,
DT_ThanhTien) VALUES ('DT00002', 'KH00002', 'DS00002', 2, '2022-10-04',
NULL);
```

```
INSERT INTO BT_DonThuoc (DT_Ma, KH_Ma, DS_Ma, DT_SoLuong, DT_Ngay,
DT_ThanhTien) VALUES ('DT00003', 'KH00003', 'DS00002', 1, '2022-10-05',
NULL);
```

```
INSERT INTO BT_DonThuoc (DT_Ma, KH_Ma, DS_Ma, DT_SoLuong, DT_Ngay,
DT_ThanhTien) VALUES ('DT00004', 'KH00004', 'DS00003', 2, '2022-10-05',
NULL);
```

```
INSERT INTO BT_DonThuoc (DT_Ma, KH_Ma, DS_Ma, DT_SoLuong, DT_Ngay,
DT_ThanhTien) VALUES ('DT00005', 'KH00005', 'DS00003', 3, '2022-10-06',
NULL);
```

```
select * from BT_DonThuoc;
```

commit;

-- Chi Tiet Don Thuoc

INSERT INTO BT_ChiTietDonThuoc (DT_Ma, DP_Ma, CTDT_SoLuong, CTDT_Gia) VALUES ('DT00001', 'DP00001', 1, NULL);

INSERT INTO BT_ChiTietDonThuoc (DT_Ma, DP_Ma, CTDT_SoLuong, CTDT_Gia) VALUES ('DT00002', 'DP00001', 2, NULL);

INSERT INTO BT_ChiTietDonThuoc (DT_Ma, DP_Ma, CTDT_SoLuong, CTDT_Gia) VALUES ('DT00002', 'DP00002', 1, NULL);

INSERT INTO BT_ChiTietDonThuoc (DT_Ma, DP_Ma, CTDT_SoLuong, CTDT_Gia) VALUES ('DT00003', 'DP00002', 2, NULL);

INSERT INTO BT_ChiTietDonThuoc (DT_Ma, DP_Ma, CTDT_SoLuong, CTDT_Gia) VALUES ('DT00003', 'DP00003', 3, NULL);

select * from BT_ChiTietDonThuoc;

commit;

select * from BT_KhachHang;

3.2. Thủ tục

-- Thu tục

3.2.1. Thủ tục thêm 1 dược phẩm mới

--- Thu tục 1. Them 1 duoc pham moi Insert_DuocPham -----

```
CREATE OR REPLACE PROCEDURE Insert_DuocPham(
    p_dp_ma IN BT_DuocPham.dp_ma%TYPE,
    p_dm_ma IN BT_DuocPham.dm_ma%TYPE,
    p_dp_ten IN BT_DuocPham.dp_ten%TYPE,
    p_dp_mota IN BT_DuocPham.dp_mota%TYPE,
    p_dp_gia IN BT_DuocPham.dp_gia%TYPE)
```

IS

BEGIN

```
INSERT INTO BT_DuocPham (DP_Ma, DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
VALUES (p_dp_ma, p_dm_ma, p_dp_ten, p_dp_mota, p_dp_gia);
COMMIT;
END;
```

--- Thuc thi thu tuc -----

```
select * from BT_DuocPham;
```

```
EXECUTE Insert_DuocPham ('DP00006', 'DM00003', 'Akitykity-New', 'Nhiem mot
loai hoac nhieu loai ki sinh trung duong ruot nhu giun dua, giun kim, san hat dua,...',
10000);
```

3.2.2. Thủ tục cập nhật giá của 1 dược phẩm

--- Thu tuc 2. Cap nhat gia cua 1 duoc pham Update_GiaDuocPham -----

```
CREATE OR REPLACE PROCEDURE Update_GiaDuocPham(
    p_dp_ma IN BT_DuocPham.dp_ma%TYPE,
    p_dp_gia IN BT_DuocPham.dp_gia%TYPE
)
```

IS

BEGIN

```
UPDATE BT_DuocPham SET dp_gia = p_dp_gia where DP_Ma = p_dp_ma;
COMMIT;
```

END;

commit;

--- Thuc thi thu tuc -----

```
EXECUTE Update_GiaDuocPham('DP00006', 100000);
```

```
select * from BT_DuocPham;
```

commit;

3.2.3. Thủ tục xóa 1 dược phẩm

--- Thu tục 3. Xoa 1 duoc pham Delete_DuocPham -----

CREATE OR REPLACE PROCEDURE Delete_DuocPham (

p_dp_ma IN BT_DuocPham.dp_ma%TYPE

)

IS

BEGIN

DELETE BT_DuocPham WHERE DP_Ma = p_dp_ma;

COMMIT;

END;

rollback;

-- Thuc thi thu tục xoa 1 duoc pham -----

EXECUTE Delete_DuocPham ('DP00006');

select * from BT_DuocPham;

commit;

3.3. Hàm

-- Ham

3.3.1. Hàm trả về đơn giá của 1 dược phẩm

--- Ham 1. Tra ve don gia cua 1 duoc pham Get_GiaDuocPham-----

select DP_Gia from BT_DuocPham where DP_Ma = 'DP00001';

CREATE OR REPLACE FUNCTION Get_GiaDuocPham(f_dp_ma char)

RETURN varchar

IS

temp float(15);

BEGIN

```
select DP_Gia INTO temp
from BT_DuocPham
where DP_Ma = f_dp_ma;
RETURN temp;
```

END;

--- Thuc thi ham tra ve don gia cua mot duoc pham -----

```
Select Get_GiaDuocPham ('DP00001') as GiaDuocPham from dual;
select * from BT_DuocPham;
commit;
```

3.3.2. Hàm trả về ngày bán của 1 đơn thuốc nào đó

--- Ham 2. Tra ve ngay ban cua mot don thuoc nao do-----

-- select tra ve nhung don thuoc co chua 1 duoc pham cu the nao do

```
select DT_Ma
from BT_ChiTietDonThuoc c JOIN BT_DuocPham d ON c.DP_Ma = d.DP_Ma
where d.DP_Ma = 'DP00001';
select DT_Ngay
from BT_DonThuoc where DT_Ma = 'DT00003';
```

CREATE OR REPLACE FUNCTION Get_NgayDonThuoc(f_dt_ma char)

RETURN varchar

IS

temp date;

BEGIN

```
select DT_Ngay INTO temp
from BT_DonThuoc where DT_Ma = f_dt_ma;
RETURN temp;
```

END;

drop function get_donthuoc;

--- Thuc thi ham tra ve ngay ban cua 1 don thuoc -----

Select Get_NgayDonThuoc('DT00001') as NgayBanCuaDonThuoc from dual;

commit;

3.3.3. Hàm trả về thành tiền của 1 đơn thuốc nào đó

--- Ham 3. Tra ve thanh tien cua 1 don thuoc nao do Get_ThanhTienDonThuoc----

--select * from BT_DuocPham;

--select * from BT_ChiTietDonThuoc;

--select * from BT_DonThuoc;

select DP_Gia * CTDT_SoLuong * DT_SoLuong

from BT_DuocPham d JOIN BT_ChiTietDonThuoc c ON d.DP_Ma = c.DP_Ma

JOIN BT_DonThuoc e ON e.DT_Ma = c.DT_Ma

where e.DT_Ma = 'DT00001';

CREATE OR REPLACE FUNCTION Get_ThanhTienDonThuoc(f_dt_ma char)

RETURN varchar

IS

temp float(15);

BEGIN

select (DP_Gia * CTDT_SoLuong * DT_SoLuong) INTO temp

from BT_DuocPham d JOIN BT_ChiTietDonThuoc c ON d.DP_Ma = c.DP_Ma

JOIN BT_DonThuoc e ON e.DT_Ma = c.DT_Ma

where e.DT_Ma = f_dt_ma;

RETURN temp;

END;

--- Thuc thi ham tra ve thanh tien cua 1 don thuoc nao do -----

```
Select Get_ThanhTienDonThuoc ('DT00001') as ThanhTien from dual;
commit;
```

3.4. Trigger

-- Trigger

3.4.1. Trigger theo dõi thêm dược phẩm mới

--- Trigger 1. Theo doi them duoc pham moi -----

--- Tao bang luu thong tin khi du lieu duoc sua doi

```
CREATE TABLE insert_NewDuocPham(
```

```
    UserName VARCHAR2(20),
```

```
    MODIFY_TIME DATE,
```

```
    New_DP_Ma char(7),
```

```
    New_DM_Ma char(7),
```

```
    New_DP_Ten varchar(200),
```

```
    New_DP_MoTa varchar(300),
```

```
    New_DP_Gia float(15)
```

```
);
```

-- Tao trigger theo doi them duoc pham moi -----

```
CREATE OR REPLACE TRIGGER store_insert_NewDuocPham
```

```
AFTER INSERT ON BT_DuocPham
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO Insert_NewDuocPham(UserName,    MODIFY_TIME,
New_DP_Ma, New_DM_Ma, New_DP_Ten, New_DP_MoTa, New_DP_Gia)
```

```
VALUES (User,
Sysdate,:new.DP_Ma,:new.DM_Ma,:new.DP_Ten,:new.DP_MoTa,:new.DP_Gia);
END;
commit;
```

-- Them 1 duoc pham moi dekiem tra thu trigger

```
EXECUTE Insert_DuocPham ('DP00006', 'DM00003', 'Akitykity-New', 'Nhiem mot
loai hoac nhieu loai ki sinh trung duong ruot nhu giun dua, giun kim, san hat dua,...',
10000);
```

```
select * from insert_newduocpham;
```

```
commit;
```

3.4.2. Trigger theo dõi sửa giá dược phẩm

--- Trigger 2. Theo doi sua gia duoc pham -----

--- Tao bang luu thong tin khi du lieu duoc sua doi

```
CREATE TABLE Change_GiaDuocPham(
```

```
  UserName VARCHAR2(20),
```

```
  MODIFY_TIME DATE,
```

```
  DP_Ma char(7),
```

```
  Old_Gia float(15),
```

```
  New_Gia float(15)
```

```
);
```

-- Tao trigger theo doi sua gia trong duoc pham -----

```
CREATE OR REPLACE TRIGGER store_change_GiaDuocPham
```

```
AFTER UPDATE OF DP_Gia on BT_DuocPham
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO Change_GiaDuocPham(UserName, MODIFY_TIME, DP_Ma,
Old_Gia, New_Gia)
VALUES (User, Sysdate,:new.DP_Ma,:old.DP_Gia,:new.DP_Gia);
END;
commit;
```

-- Cap nhat gia duoc pham de kiem tra thu trigger

```
EXECUTE Update_GiaDuocPham('DP00006', 100000);
select * from change_giaduocpham;
commit;
```

3.4.3. Trigger theo dõi xóa dược phẩm

--- Trigger 3. Theo doi xoa du lieu trong duoc pham -----

--- Tao bang luu thong tin khi du lieu duoc sua doi

```
CREATE TABLE trigger_delete_DuocPham(
    UserName VARCHAR2(20),
    MODIFY_TIME DATE,
    DP_Ma char(7),
    DM_Ma char(7),
    DP_Ten varchar(200),
    DP_MoTa varchar(300),
    DP_Gia float(15)
);
```

-- Tao trigger theo doi xoa duoc pham -----

```
CREATE OR REPLACE TRIGGER store_delete_DuocPham
AFTER DELETE ON BT_DuocPham
FOR EACH ROW
BEGIN
    INSERT INTO trigger_delete_DuocPham(Username, MODIFY_TIME, DP_Ma,
DM_Ma, DP_Ten, DP_MoTa, DP_Gia)
        VALUES                                     (User,
Sysdate,:old.DP_Ma,:old.DM_Ma,:old.DP_Ten,:old.DP_MoTa,:old.DP_Gia);
END;
commit;

-- Xoa duoc pham de kiem tra thu trigger

EXECUTE Delete_DuocPham ('DP00006');
select * from trigger_delete_duocpham;
commit;

ROLLBACK;
commit;--
--
```