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](#_Toc118988548)

[1.1. HQTCSDL là gì? Các HQTCSDL hiện nay 1](#_Toc118988549)

[1.2. Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì? 1](#_Toc118988550)

[1.3. DBA là ai? Nhiệm vụ? 2](#_Toc118988551)

[1.4. Các phương pháp bảo vệ dữ liệu 3](#_Toc118988552)

[1.5. Giao dịch là gì? Tại sao cần phải giao dịch? 3](#_Toc118988553)

[1.6. Các trạng thái của giao dịch 4](#_Toc118988554)

[1.7. Thuộc tính ACID là gì? 4](#_Toc118988555)

[1.8. Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục? 5](#_Toc118988556)

[1.9. Lịch trình là gì? Lịch trình tuần tự và không tuần tự 5](#_Toc118988557)

[1.10. Các kỹ thuật quản lý cạnh tranh 5](#_Toc118988558)

[1.10.1. Bi quan (lock) 5](#_Toc118988559)

[1.10.2. Lạc quan: 3 kỳ (đọc, kiểm tra, ghi) 6](#_Toc118988560)

[1.10.3. Độ mịn của mục dữ liệu 7](#_Toc118988561)

[1.11. Phục hồi dữ liệu 7](#_Toc118988562)

[1.11.1. Tại sao cần phục hồi dữ liệu 7](#_Toc118988563)

[1.11.2. Điểm kiểm tra là gì (check point) 7](#_Toc118988564)

[1.11.3. Các kỹ thuật phục hồi dữ liệu 7](#_Toc118988565)

[II. BÀI TẬP 1 – CSDL XE Ô TÔ 9](#_Toc118988566)

[2.1. Câu 1: Tạo các bảng để lưu thông tin 9](#_Toc118988567)

[2.2. Câu 2: Viết thủ tục cho phép nhập vào 1 xe mới 22](#_Toc118988568)

[2.3. Câu 3: Gọi thủ tục để Nhập liệu như bảng trên 23](#_Toc118988569)

[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](#_Toc118988570)

[2.5. Câu 5: Viết hàm trả về giá đàm phán của 1 xe nào đó 24](#_Toc118988571)

[2.6. Câu 6: Viết hàm trả về loại xe, hãng xe có giá cao nhất 26](#_Toc118988572)

[2.7. Câu 7: Viết thủ tục để xóa 1 xe nào đó 28](#_Toc118988573)

[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](#_Toc118988574)

[2.9. Câu 9: Viết trigger để theo dõi việc sửa giá trong bảng xe 30](#_Toc118988575)

[2.10. Câu 10: Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe 31](#_Toc118988576)

[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](#_Toc118988577)

[2.12. Câu 12: Tạo người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng 32](#_Toc118988578)

[III. BÀI TẬP 2 – PHARMACY DATABASE 34](#_Toc118988579)

[3.1. Mô tả và thiết kế CSDL về một lĩnh vực quản lý hiệu thuốc 34](#_Toc118988580)

[3.1.1. Tạo bảng 34](#_Toc118988581)

[3.1.2. Nhập dữ liệu cho bảng 37](#_Toc118988582)

[3.2. Thủ tục 40](#_Toc118988583)

[3.2.1. Thủ tục thêm 1 dược phẩm mới 40](#_Toc118988584)

[3.2.2. Thủ tục cập nhật giá của 1 dược phẩm 41](#_Toc118988585)

[3.2.3. Thủ tục xóa 1 dược phẩm 42](#_Toc118988586)

[3.3. Hàm 42](#_Toc118988587)

[3.3.1. Hàm trả về đơn giá của 1 dược phẩm 42](#_Toc118988588)

[3.3.2. Hàm trả về ngày bán của 1 đơn thuốc nào đó 43](#_Toc118988589)

[3.3.3. Hàm trả về thành tiền của 1 đơn thuốc nào đó 44](#_Toc118988590)

[3.4. Trigger 45](#_Toc118988591)

[3.4.1. Trigger theo dõi thêm dược phẩm mới 45](#_Toc118988592)

[3.4.2. Trigger theo dõi sửa giá dược phẩm 46](#_Toc118988593)

[3.4.3. Trigger theo dõi xóa dược phẩm 47](#_Toc118988594)

**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 |

# TÓM TẮT LÝ THUYẾT

## 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)

## 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/*](https://www.geeksforgeeks.org/difference-between-dbms-and-spreadsheet/)*)*

## 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:
  + 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 HQTCSDL 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)*

## 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)*

## 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)*

## 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)*

## 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)*

## 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:
  + *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.
  + *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.
  + *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.
* 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)*

## 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)*

## Các kỹ thuật quản lý cạnh tranh

### 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.
  + *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.
  + *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)*

### 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)*

### 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 l ý nơi lưu trữ các quan hệ
* Một mẫu tin (record)
* Một giá trị của một trường trong một mẫu tin.

*(Chương 3 – Trang 37)*

## Phục hồi dữ liệu

### 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)*

### 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)*

### 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)*

# 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;

## 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;

-- Thong So Ky Thuat

create table ThongSoKyThuat(

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 ThongSoKyThuat MODIFY TSKT\_MomenXoan varchar(200);

-- Noi Lan Banh

create table NoiLanBanh(

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 NoiLanBanh

ADD NLB\_GiaTri float(15); -- chi so % nhan voi gia niem yet

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 xuat ngan hang 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', 1839000000);

INSERT INTO Xe (X\_Ma, PB\_Ma, PKX\_Ma, DC\_Ma, DP\_Ma, X\_GiaNiemYet) VALUES ('X000003', 'PB00010','PKX0004', 'DC00007', NULL, 3600000000);

INSERT INTO Xe (X\_Ma, PB\_Ma, PKX\_Ma, DC\_Ma, DP\_Ma, X\_GiaNiemYet) VALUES ('X000004', 'PB00022','PKX0005', 'DC00009', 'DP00004', 425000000);

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 ThongSoKyThuat VALUES ('TSKT004', 'X000004','98/6200', '128/4400', 'Tu ding vo cap - CVT', 'Cau truoc - FWD', 5);

commit;

delete from ThongSoKyThuat;

select \* from ThongSoKyThuat;

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

-- 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;

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

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

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

-- Cau 2 Viet thu tuc cho phep 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\_pkx\_ma IN Xe.pkx\_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\_pkx\_ma, p\_dc\_ma, p\_dp\_ma, p\_x\_gianiemyet);

COMMIT;

END;

## 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', 18000000000);

commit;

## 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;

## 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;

## 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;\*/

## 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;

## 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;

## 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';

## 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\_GiaNiemYet);

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';

## 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;

## 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;--

--

# BÀI TẬP 2 – PHARMACY DATABASE

-- Le Thai Giao Tien B1906586 --

select sysdate from dual;

-- Mo ta va thiet ke CSDL ve mot linh vuc nao do (ban hang, ql sinh vien, quan ly quan cafe, du lich,...) gom it nhat 5 bang.

-- Moi cot trong bang phai co cac rang buoc (Khoa chinh, khoa ngoai, RB duy nhat, RB kiem tra, not null, mac dinh, bieu thuc chinh quy - vd Ma hang: MH01,...)

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

-- Ten bang (cac doi tuong khac) bat dau la "BT\_..."

-- Viet it nhat 3 ham, 3 thu tuc, 3 trigger de quan y du lieu tren

--

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;

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

-- Mo ta va thiet ke CSDL ve mot linh vuc quan ly hieu thuoc

### Tạo bảng

-- Khach Hang

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}')),-- Bat dau so 0 tiep theo la 9 hoac 10 so

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;--

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

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

### Nhập dữ liệu cho bảng

-- Nhap 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', 'Giam ho, giam viem mui di ung va viem phe quan di ung', 26500);

INSERT INTO BT\_DuocPham (DP\_Ma, DM\_Ma, DP\_Ten, DP\_MoTa, DP\_Gia) VALUES ('DP00003', 'DM00002', 'Avircrem', 'Cac truong hop nhiem virus Herpes simplex tren da', 25000);

INSERT INTO BT\_DuocPham (DP\_Ma, DM\_Ma, DP\_Ten, DP\_MoTa, DP\_Gia) VALUES ('DP00004', 'DM00002', 'Axcel Fungicort Cream', 'Dieu tri viem da do nam, ton thuong do con trung dot', 32000);

INSERT INTO BT\_DuocPham (DP\_Ma, DM\_Ma, DP\_Ten, DP\_MoTa, DP\_Gia) VALUES ('DP00005', 'DM00003', 'Agimoti Agimexpharm', 'Buon non va non nang, cam giac chuong va nang vung thuong vi', 32000);

select \* from BT\_DuocPham;

commit;

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

-- Don thuoc

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;

## Thủ tục

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

-- Thu tuc

### Thủ tục thêm 1 dược phẩm mới

--- Thu tuc 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);

### 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;

### Thủ tục xóa 1 dược phẩm

--- Thu tuc 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 tuc xoa 1 duoc pham --------------------------------------------------------------------------------------------------------

EXECUTE Delete\_DuocPham ('DP00006');

select \* from BT\_DuocPham;

commit;

## Hàm

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

-- Ham

### 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;

### 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;

### 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;

## Trigger

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

-- Trigger

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

### 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 de kiem 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;

### 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;

### 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;--

--