**BÀI TẬP THỰC HÀNH SQL**

**Nội dung:**

Phần 1. Ngôn ngữ SQL

Phần 2. Ràng buộc dữ liệu

Phần 3. Trigger (Tham khảo)

**PHẦN 1: NGÔN NGỮ SQL**

**Mục đích:**

Thực hành các yêu cầu truy vấn Cơ sở dữ liệu sử dụng ngôn ngữ SQL.

***Bài 1***

1. Tạo một cơ sở dữ liệu (CSDL) dùng để quản lý nhân viên, lấy tên CSDL là  QLNhanVien.

2. Cấu trúc của CSDL gồm các bảng với các quan hệ sau:

Diagram

Description automatically generated

a. NHANVIEN

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **MaNV** | Ký tự | 8 | Chữ hoa +số |
| HoTen | Ký tự | 30 |  |
| Congviec | Ký tự | 50 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Luong | Số |  |  |
| MaDV | Ký tự | 4 |  |
| MaPT | Ký tự | 8 | Chữ hoa + số  (Mã người phụ trách) |

b. DONVI

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **MaDV** | Ký tự | 4 | Chữ hoa +số |
| TenDV | Ký tự | 50 |  |

c. BACLUONG

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **MaBac** | Ký tự | 50 | Chữ hoa +số |
| BacCao | Số |  |  |
| BacThap | Số |  |  |

Chú ý: Mỗi câu lệnh tạo bảng được viết bằng 1 Query, đặt tên lần lượt là 21, 2b, 2c. 3. Thêm dữ liệu (bằng lệnh INSERT) cho bảng NHANVIEN để có kết quả như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NHANVIEN** | | | | |
| **MaNV** | **HoTen** | **CongViec** | **Luong** | **MaDV** |
| NV001 | Phạm Thị Nhàn | Thư ký |  | 500 0001 |
| NV002 | Hoàng Thanh Vân | Giáo viên |  | 600 0001 |
| NV003 | Hoàng Thị Lan | Giáo viên |  | 200 0002 |
| NV004 | Đỗ Trung Dũng | Thư ký |  | 700 0003 |

4. Thêm dữ liệu (bằng lệnh INSERT) cho bảng DONVI để có kết quả như sau:

|  |  |
| --- | --- |
| **DONVI** | |
| **MaDV** | **TenDV** |
| 0001 | KHTN |
| 0002 | DHTL |
| 0003 | DHQG |

5. Thêm dữ liệu (bằng lệnh INSERT) cho bảng BACLUONG để có kết quả như sau:

|  |  |  |
| --- | --- | --- |
| **BACLUONG** | | |
| **MaBac** | **BacThap** | **BacCao** |

|  |  |  |
| --- | --- | --- |
| **BACLUONG** | | |
| **MaBac** | **BacThap** | **BacCao** |
| 1 | 400 | 500 |
| 2 | 501 | 600 |
| 3 | 601 | 800 |

6. Dùng câu lệnh truy vấn dữ liệu, đưa ra tất cả thông tin về nhân viên. 7. Đưa ra Hoten, Congviec, Luong của tất cả các nhân viên.

8. Đưa ra toàn bộ công việc của các nhân viên, các giá trị không trùng nhau. 9. Đưa ra Hoten, LuongQuy của tất cả các nhân viên, với LuongQuy= Luong\*3. 10.Đưa ra Hoten, Luong sắp xếp theo thứ tự tăng dần/ giảm dần của Luong. 11.Đưa ra Hoten, Luong của các nhân viên có Luong>300.

12.Đưa ra Hoten, Luong của các nhân viên có Luong>300 và làm công việc là Giáo viên. 13.Đưa ra những nhân viên có lương hoặc 200, 300, 600.

14.Đưa ra những nhân viên có Lương trong khoảng 300 đến 600.

15.Đưa ra Hoten, Congviec của các nhân viên có Họ tên bắt đầu bằng chữ ‘Hoàng’.

16.Đưa ra lương trung bình, lương lớn nhất, nhỏ nhất của tất cả các nhân viên trong bảng  NHANVIEN.

17.Đưa ra Côngviệc, Lương trung bình của từng loại công việc.

18.Đưa ra Côngviệc, Lương trung bình của tất cả các nhân viên có Luong>200 theo từng  loại công việc.

19.Đưa ra tổng lương của từng nhóm công việc trong từng đơn vị.

Select madonvi, congviec, sum(luong) from nhan vien nv group by congviec, madonvi

20.Đưa ra những Congviec và trung bình lương của các công việc có trung bình lương  >=300

21.Đưa ra những đơn vị và lương lớn nhất của các đơn vị có lương lớn nhất >=300. 22.Đưa ra Hoten, Congviec, TenDV của tất cả nhân viên.

23.Câu lệnh sau cho kết quả như thế nào.

SELECT HoTen, CongViec, TenDV

FROM NHANVIEN NV, DONVI DV

WHERE NV.MaDV= DV.MaDV;

24.Đưa ra HoTen, Congviec, TenDV, Luong của những nhân viên có Luong>=500. 25.Câu lệnh sau cho kết quả như thế nào.

SELECT NV.MaNV, NV.Hoten, PT.MaNV, PT.Hoten

FROM NHANVIEN NV, NHANVIEN PT

WHERE (NV.MaNV=PT.MaPT) AND (NV.Luong>PT.Luong);

27.Đưa ra những công việc trong đơn vị 1 có MaDV là 0001 và đơn vị 2 có MaDV là  0002.

28.Đưa ra Hoten, TenDV, Congviec, Luong của những người có lương lớn hơn lương  trung bình của toàn bộ nhân viên.

Select ho ten, tendonvi, congviec,luong from nhanvien nv, donvi dv where  (nv.madv=dv.madv) and (luong > select avg(luong) from nhanvien)

29.Đưa ra những nhân viên có lương lớn hơn người có lương lớn nhất trong đơn vị có tên  là DHTL.

Select hoten from nhanvien where (luong > select max(luong) from nhanvien, donvi  where (nhanvien.madv=donvi.madonvi) and (donvi,tendonvi=’DHTL))

30.Đưa ra Hoten, MaDV, Luong của các nhân viên có Luong=Luong thấp nhất trong đơn  vị của họ.

31.Đưa ra Hoten, MaDV, Luong của các nhân viên có Luong=Luong thấp nhất trong một  đơn vị nào đó.

32.Đưa ra Hoten, Luong của các nhân viên có Luong lớn nhất của đơn vị có mã đơn vị là  0002.

33.Đưa ra MaDV, AVG(Luong) của đơn vị có trung bình lương lớn hơn lương nhỏ nhất  của đơn vị có mã đơn vị là 0003.

***Bài 2***

Thực hiện các thao tác sau cho Cơ sở dữ liệu dùng để quản lý một cửa hàng kinh doanh. I. **Định nghĩa cơ sở dữ liệu**

1. **Tạo cơ sở dữ liệu**

Tạo một cơ sở dữ liệu dùng để quản lý cửa hàng đặt tên là QLCH bao gồm các bảng có mối quan  hệ như sau:

Diagram

Description automatically generated

**Cấu trúc của mỗi bảng như sau:**

1. KHACH

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **Ma\_khach** | Ký tự | 6 | Chữ hoa +số |
| Ten\_kh | Ký tự | 30 | Chữ đầu viết hoa |
| Diachi\_kh | Ký tự | 30 | Chữ đầu viết hoa |

2. HANG

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **Ma\_hang** | Ký tự | 6 | Chữ hoa+số |
| Ten\_hang | Ký tự | 15 | Chữ đầu viết hoa |
| Mota\_hang | Ký tự | 30 | Chữ đầu viết hoa |

|  |  |  |  |
| --- | --- | --- | --- |
| Don\_vi | Ký tự | 10 | Chữ thường |

3. DONHANG

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **So\_don** | Ký tự | 6 | Chữ hoa +số |
| Ma\_khach | Ký tự | 15 | Chữ hoa +số |
| Ngay\_don | Ngày | 8 | Dd/mm/yy |

4. DONGDON

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **So\_don** | Ký tự | 6 | Chữ hoa +số |
| **Ma\_hang** | Ký tự | 6 | Chữ hoa +số |
| So\_luongd | Số | 7 | Số nguyên |

5. PHIEUGIAO

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **So\_phieu** | Ký tự | 6 | Chữ hoa +số |
| Ma\_khach | Ký tự | 6 | Chữ hoa +số |
| Ngay\_giao | Ngày | 30 | Dd/mm/yy |
| Noi\_giao | Ký tự | 30 |  |
| Tong\_tien | Số | 9 | Số thực |

6. DONGPHIEU

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Kích cỡ** | **Khuôn dạng** |
| **So\_phieu** | Ký tự | 6 | Chữ hoa+ số |
| **Ma\_hang** | Ký tự | 6 | Chữ hoa + số |
| Don\_gia | Số | 6 | Số thực |
| So\_luonggi | Số | 7 | Số thực |

**2. Sửa đổi cấu trúc:**

1. Thêm một thuộc tính:

- Trong bảng **KHACH**, thêm một thuộc tính ***SoThich*** nhằm lưu vào bảng KHACH sở  thích của khách hàng.

- Trong bảng **HANG**, thêm hai thuộc tính là ***NoiCungCap*** và ***NoiSanXuat*** nhằm lưu  vào bảng HANG thông tin về nơi cung cấp hàng hóa đó và nơi sản xuất chúng.

2. Thay đổi kiểu của một thuộc tính.

- Trong bảng KHACH, thay đổi kiểu dữ liệu cho trường SoThich, từ kiểu Text thành  kiểu Memo.

- Trong bang HANG, thay đổi kiểu dữ liệu cho trường NoiCungCap, từ kiểu Text  thành kiểu Number.

3. Xóa một thuộc tính.

- Trong bảng HANG, hãy xóa 2 thuộc tính NoiCungCap và NoiSanXuat. - Trong bảng KHACH, hãy xóa thuộc tính SoThich.

**II. Quản trị cơ sở dữ liệu**

**1. Thêm giá trị vào các dòng của bảng**

- Thêm vào bảng KHACH các giá trị như sau:

|  |  |  |
| --- | --- | --- |
| **KHACH** | | |
| **Ma\_khach** | **Ten\_kh** | **diachi\_kh** |
| K001 | Đào Minh Thư | Đại học Quốc Gia |
| K002 | Nguyễn Liên Dung | BNC |
| K003 | Phạm Hoàng Nhung | Đại học Thủy Lợi |

- Thêm vào bảng DONHANG các giá trị như sau:

|  |  |  |
| --- | --- | --- |
| **DONHANG** | | |
| **so\_don** | **Ma\_khach** | **ngay\_don** |
| DH001 | K001 | 3/22/2002 |
| DH002 | K001 | 7/13/2003 |
| DH003 | K002 | 12/24/2002 |

- Thêm vào tất cả các bảng, mỗi bảng một số bộ giá trị phù hợp.

***Chú ý***: Hãy thử trường hợp thêm ma\_khach là K0010 vào bảng DONHANG thì kết quả thế  nào? Tại sao?

**2. Thêm giá trị cho một số thuộc tính**

- Thêm giá trị vào bảng KHACH để được giá trị như sau:

|  |  |  |
| --- | --- | --- |
| **KHACH** | | |
| **Ma\_khach** | **Ten\_kh** | **Diachi\_kh** |
| K001 | Đào Minh Thư | Đại học Quốc Gia |
| K002 | Nguyễn Liên Dung | BNC |
| K003 | Phạm Hoàng Nhung | Đại học Thủy Lợi |
| K004 | Trịnh Hồng Cường |  |
| K005 | Nguyễn Văn Hải | Đại học Thủy Lợi |

- Thêm giá trị vào bảng HANG để được giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **HANG** | | | |
| **ma\_hang** | **ten\_hang** | **Mota\_hang** | **don\_vi** |
| H001 | Hoa loa kèn | Hoa trắng, to | Bông |
| H002 | Hoa hồng | Nhiều mau | Bông |
| H003 | Hoa lan |  | Cành |

**3. Xóa hàng**

- Xóa một hàng có ma\_hang = ‘H001’ trong bảng HANG.

- Xóa tất cả các khách hàng có Diachi\_kh = ‘Đại học Thủy Lợi ‘ trong bảng KHACH.  - Xóa khách hàng có tên là Trịnh Hồng Cường.

**4. Sửa đổi giá trị của một hàng.**

- Sửa đổi địa chỉ của khách hàng thành địa chỉ mới là Đại học Quốc Gia với  ma\_khach là K002.

- Sửa đổi tất cả các tên hàng là “Hoa lan” thành “Hoa phong lan” và mô tả hàng là  “Hàng nhập khẩu từ Đà Lạt”

**III. Các lệnh truy vấn dữ liệu**

1. Xem toàn bộ nội dung của bảng KHACH

2. Xem toàn bộ nội dung của bảng HANG.

3. Xem toàn bộ nội dung của bảng DONHANG.

4. Đưa ra Tên và Địa chỉ của tất cả các Khách hàng trong bảng KHACH.

5. Đưa ra Tên hàng, Mô tả hàng và Đơn vị tính của tất cả các mặt hàng trong bảng  HANG.

6. Đưa ra tất cả các đơn vị tính dùng để tính hàng hóa.

7. Đưa ra tất cả các tên hàng trong bảng HANG.

8. Đưa ra tất cả các địa chỉ của khách hàng.

9. Đưa ra so\_phieu, ma\_hang, don\_gia, soluonggi và Thành tiền (don\_gia\*so\_luong)  của tất cả các hàng trong bảng PHIEU\_GIAO.

10. Đưa ra so\_phieu, ma\_hang, don\_gia, soluonggi và Thành tiền (don\_gia\*so\_luong),  sử dụng bí danh la thanh\_tien của tất cả các hàng trong bảng PHIEU\_GIAO .

11. Đưa ra tất cả giá trị của bảng HANG theo thứ tự giảm dần của ma\_hang.

12. Đưa ra tất cả các giá trị của bảng PHIEUGIAO theo thứ tự tăng dần (giảm dần) của  tổng tiền (tong\_tien).

13. Đưa ra tất cả các khách hàng có địa chỉ là Đại học Quốc Gia trong bảng KHACH.

14. Đưa ra So\_phieu, Ma\_khach trong bảng PHIEUGIAO với điều kiện Tong\_ tien>=100.000, và sắp xếp theo điều kiện giảm dần của Tong\_tien.

15. Đưa ra ten\_hang, mota\_hang của những hàng hóa có don\_vi được tính theo bông  trong bảng HANG.

16. Đưa ra danh sách các bản ghi bao gồm so\_phieu, ma\_khach, ngay\_giao trong bảng  PHIEUGIAO với điều kiện Noi\_giao là Hà Nội và Tong\_tien>50.000.

17. Đưa ra danh sách các bản ghi bao gồm Ma\_khach, Ten\_khach trong bảng KHACH  với điều kiện địa chỉ của khách ở Đại học Quốc Gia hoặc Đại học Thủy Lợi.

18. Đưa ra những mặt hàng trong bảng HANG có đơn vị tính là Bông, Cành hoặc Bó.

19. Đưa ra danh sách những khách hàng (Ma\_khach, Tong \_tien) nằm trong bảng PHIEUGIAO có Noi\_giao nằm trong số các địa điểm sau:  Hà Nội, Hồ Chí Minh, Hải Phòng.

20. Đưa ra tất cả những thông tin trong bảng PHIEUGIAO với điều kiện tong\_tien nằm  trong khoảng từ 100.000 đến 500.000.

21. Đưa ra ma\_kh, ten\_kh trong bảng KHACH của những khách hàng có họ là Nguyễn.

22. Đếm số lần mua hàng của khách hàng có ma\_kh là K001 trong bảng PHIEUGIAO.  Thuộc tính mới này được đặt tên là SoLanMua.

23. Tính tổng tiền trung bình của mỗi PHIEUGIAO. Thuộc tính mới này được đặt tên là  TrungBinhPG.

24. Tính tổng tiền trong bảng PHIEUGIAO của những khách hàng có ma\_khach =  ‘K002’.

25. Tính tổng số các khách hàng trong bảng KHACH có địa chỉ (diachi\_KH) là Đại học  Thủy Lợi.

26. Đưa ra ma\_khach, ngay\_giao, noi\_giao trong bảng PHIEUGIAO có tong\_tien thanh  toán lớn nhất.

27. Đưa ra ma\_khach, ngay\_giao, noi\_giao trong bảng PHIEUGIAO có tong\_tien thanh  toán nhỏ nhất.

**PHẦN 2. RÀNG BUỘC DỮ LIỆU**

**Mục đích:**

- Sử dụng Enterprise Manager để thực hiện các Constraints và Rules - Sử dụng Enterprise Manager Wizard để tạo các chỉ số

- Sử dụng QA để thêm các constraints

- Sử dụng QA để tạo các rules và gán các rules cho các đối tượng

- Sử dụng QA để tạo và xem chỉ số

***1. Hướng dẫn trực tiếp***

**1.1 Tạo ràng buộc PRIMARY KEY**

Các bước thực hiện:

- Vào Enterprise Manager

- Vào phần thiết kế của bảng muốn tạo khoá chính

- Bôi đen một trường hoặc nhiều trường làm khoá chính

- Kích vào biểu tưởng Khoá trên thanh công cụ

Graphical user interface, table

Description automatically generatedHình 11.1

**1.2 Tạo ràng buộc Unique**

Ví dụ tạo ràng buộc Unique cho cột PNR\_No của bảng Passenger. Các bước thực hiện:

1. Chọn phần thiết kế của bảng Passenger

2. Chọn cột PNR\_No, kích phải chuột, chọn Properties.

3. Chọn thẻ Indexes/Keys

4. Kích nút New

5. Lựa chọn như hình sau

Graphical user interface, application

Description automatically generated

Hình 11.2

6. Kích Close

7. Kích Save để ghi lại thiết kế bảng

**1.3 Sử dụng thuộc tính IDENTITY**

Ví dụ quy định cột PNR\_No của bảng Passenger là IDENTITY.

Các bước thực hiện:

1. Chọn phần thiết kế bảng Passenger

2. Chọn cột PNR\_No

3. Trong thẻ Columns bên dưới, chọn thuộc tính Identity = Yes

4. Chọn giá trị khởi đầu trong mục Identity Seed, giả sử=5

5. Chọn giá trị tăng trong mục Identity Increment, giả sử=1 (tăng với chỉ số tăng  bằng 1)

6. Kích Save để ghi lại thiết kế bảng

**1.4 Tạo ràng buộc Default**

Giả sử đặt giá trị mặc định cho cột Status trong bảng Reservation = ‘R’ Các bước thực hiện:

1. Chọn cơ sở dữ liệu Flight Information

2. Kích phải chuột vào đối tượng Default, chọn New Default…  3. Nhập tên của ràng buộc Default vào mục Name

4. Chọn giá trị Default vào mục Value

Graphical user interface, application

Description automatically generated

Hình 11.3

5. Kích phải chuột vào đối tượng Default\_Status\_Reservation vừa được tạo, chọn  Properties

6. Kích vào Bind Columns…

7. Chọn tên bảng, tên cột chấp nhận ràng buộc này như hình sau: 8. Kích Add

9. Kích Apply

10.Kích OK

Graphical user interface, application

Description automatically generated

Hình 11.4

**1.5 Tạo ràng buộc FOREIGN KEY**

Giả sử cần thiết đặt ràng buộc cho thuộc tính Aircode trong bảng Airline\_Service là  khoá ngoại, được tham chiếu từ thuộc tính Aircode là khoá chính trong bảng  Airline\_Master. Thực hiện các bước sau:

1. Chọn cơ sở dữ liệu Flight Information

2. Chọn đối tượng Diagrams

3. Chọn New Database Diagram…

4. Kích Next

5. Chọn 2 bảng Airline\_Master và Airline\_Service. Kích Add

6. Kích Next

7. Kích Finish

Graphical user interface, application

Description automatically generatedHình 11.5

Graphical user interface, application

Description automatically generated

Hình 11.6

8. Chọn thuộc tính Aircode từ bảng Airline\_Master thả sang thuộc tính Aircode  trong bảng Airline\_Service

9. Hộp thoại Create Relationship xuất hiện, cho phép ta xác định các ràng buộc liên  quan khi thiết lập khoá ngoại

Graphical user interface, application

Description automatically generated

Hình 11.7

10.Kích OK

Diagram

Description automatically generatedHình 11.8

11. Ghi lại Diagram.

**1.6 Tạo ràng buộc Check Constraint**

Giả sử ta muốn giới hạn dữ liệu nhập vào cho cột class\_code trong bảng Reservation  chỉ nhận 3 giá trị: ‘E’, ‘Ex’, ‘FC’. Thực hiện các bước sau:

1. Chọn phần Design của bảng Reservation

2. Kích phải vào bất cứ thuộc tính nào, chọn Check Constraints

3. Kích New

4. Nhập tên của ràng buộc và giá trị như hình sau:

Graphical user interface, text, application

Description automatically generatedHình 11.9

5. Kích Close

6. Đóng và ghi lại thiết kế bảng.

**1.7 Tạo ràng buộc Not Null**

2. Chọn phần thiết kế bảng.

3. Un-check vào Allow Nulls cho cột tương ứng.

Graphical user interface, table

Description automatically generated

Hình 11.10

**1.8 Tạo Rules**

Tạo Rule để đảm bảo dữ liệu trong cột PNR\_no của bảng Reservation phải nhập  trong khoảng 1 và 500.

**Các bước thực hiện:**

1. Kích phải vào đối tượng Rules, chọn New Rule… từ menu pop-up. 2. Nhập tên của Rule, ví dụ Rule\_PNRno.

3. Soạn nội dung của Rule.

4. Kích OK.

Sau khi tạo ra Rule\_PNRno, áp dụng Rule này cho cột PNR\_no của bảng  Reservation.

Graphical user interface, application

Description automatically generated

Hình 11.11

5. Kích phải chuột vào Rule\_PNRno, và chọn Properties.

6. Kích Bind Column…

7. Chọn bảng Reservation từ Tables list.

8. Chọn cột PNR\_no từ Unbound columns list.

9. Chọn Add>>

10.Kích Apply

11.Kích OK

**Phần 3. TRIGGER (THAM KHẢO)**

**Mục đích:**

- Tạo CREATE, DELETE, UPDATE và INSTEAD OF trigger.

- Mã hoá (encrypt) trigger.

- Xem cách trigger thực hiện.

- Hiển thị các thông tin về trigger.

***1.7 Tạo INSERT trigger.***

INSERT trigger đảm bảo dữ liệu nhập vào bảng được đúng đắn.

Xem xét ví dụ tạo INSERT trigger để đảm bảo không có vé nào được đặt vào một  ngày trong quá khứ.

Các bước thực hiện:

1. Mở QA, chọn cơ sở dữ liệu FI.

2. Thực hiện đoạn lệnh sau trong QA.

|  |
| --- |
| CREATE TRIGGER insert\_trigg  ON Reservation  FOR INSERT  AS  IF((Select journey\_date From Inserted)<getdate()) BEGIN  PRINT 'journey\_date khong the nho hon ngay hien tai' ROLLBACK TRAN  END |

3. Sau đó, hãy thử thực hiện thêm một bản ghi có journey\_date < ngày hiện tại của  hệ thống.

***1.8 Tạo DELETE Trigger***

DELETE trigger ngăn cản việc xoá đi những dữ liệu quan trọng trong bảng.

Xem xét ví dụ sau: Tạo trigger để tránh xoá 2 bản ghi trong bảng Passenger đồng  thời.

1. Trong QA thực hiện như sau:

|  |
| --- |
| CREATE TRIGGER delete\_trigg  ON Passenger  FOR Delete  AS  IF((Select count(\*) From deleted)>2)  BEGIN  PRINT 'Ban khong duoc xoa cung luc 2 ban ghi'  ROLLBACK TRAN  END |

2.Thực hiện câu lệnh xoá nhiều hơn 2 bản ghi từ bảng Passenger, giả sử như sau: Kết quả:

Graphical user interface, application

Description automatically generatedHình 17.1

***1.9 Tạo UPDATE Trigger.***

**1.9.1 Tạo Table Level UPDATE Trigger.**

Trigger UPDATE sẽ được thực hiện bất cứ khi nào dữ liệu trong bảng được cập  nhật.

Xem xét ví dụ: Tạo UPDATE trigger đảm bảo rằng cột No\_of\_seats trong bảng  Reservation không được cập nhật giá trị lớn hơn 5 và journey\_date không nhỏ hơn ngày  hiện tại.

1. Thực hiện như sau trong QA.

|  |
| --- |
| CREATE TRIGGER CheckingUpdate  ON Reservation  FOR UPDATE  AS  IF((Select no\_of\_seats From inserted)>5)  OR ((Select journey\_date From Inserted)<getdate())  BEGIN  PRINT 'Error'  ROLLBACK TRAN  END |

2. Thực hiện truy vấn sau để kiểm tra Trigger:

|  |
| --- |
| UPDATE Reservation  SET no\_of\_seats=6 Where PNR\_no=3 |

Kết quả:

Graphical user interface, application, Word

Description automatically generated

Hình 17.2

**1.9.2 Tạo Column Level Update Trigger**

Loại Trigger được thực hiện khi dữ liệu trong cột nào đó được cập nhật. 1. Thực hiện như sau trong QA:

|  |
| --- |
| CREATE TRIGGER Col\_Update\_trig  ON Flight\_details  FOR UPDATE  AS  IF UPDATE(Fare)  BEGIN  PRINT 'Ban khong the update cot Fare'  ROLLBACK TRAN  END |

2. Bây giờ, chúng ta sẽ thử cập nhật cột Fare.

Hình 17.3

***1.10 Tạo Trigger có lựa chọn Encryption***

Encryption (mã hoá) là phương pháp giữ bí mật cho Trigger. Nội dung của Trigger  sau khi được mã hoá sẽ không đọc được.

1. Thực hiện câu lệnh ALTER TRIGGER để sửa insert\_trigg:

|  |
| --- |
| ALTER TRIGGER insert\_trigg  ON Reservation  WITH ENCRYPTION  FOR INSERT  AS  IF((Select journey\_date From Inserted)<getdate()) BEGIN  PRINT 'journey\_date khong the nho hon ngay hien tai' ROLLBACK TRAN  END |

3. Thực hiện câu lệnh sau:

Hình 17.4

***1.11 Hiển thị danh sách các trigger trong Database***

Sử dụng thủ tục hệ thông sp\_helptrigger để hiển thị danh sách các trigger trong cơ  sở dữ liệu hiện tại.

|  |
| --- |
| sp\_helptrigger Flight\_Details |

Kết quả:

Hình 17.5

***1.12 Sử dụng Triggers để tạo ràng buộc tham chiếu (Enforce  Referential Intergrity)***

Xem xét ví dụ sau: Tạo Trigger để kiểm tra dữ liệu nhập vào cột Meal Pref của bảng  Passenger phải là dữ liệu đã tồn tại trong cột Meal codes của bảng Meal.

1. Thực hiện như sau trong QA:

|  |
| --- |
| CREATE TRIGGER ins\_trig  ON Passenger  FOR INSERT  AS  IF (Select [Meal Pref]FROM INSERTED)  NOT IN (Select meal\_code FROM Meal)  BEGIN  Print 'Ban khong the insert gia tri nay'  ROLLBACK TRAN  END |

2. Bạn hãy thử kiểm tra hoạt động của Trigger trên.

***1.13 Cascade Delete sử dụng Nested trigger.***

Trong Nested trigger, một trigger có thể được thực hiện lồng trong trigger khác.  Chúng ta có thể lồng trigger tối đa 32 mức. Nested trigger cho phép cascade update và  cascade delete.

Thực hiện câu lệnh để kích hoạt Nested trigger:

**sp\_configure ‘nested trigger’, 1**

Ngược lại:

**sp\_configure ‘nested trigger’, 0**

Tạo Cascade delete trigger để thực hiện công việc sau: Nếu xoá một chuyến bay  trong bảng Flight, thì tất cả các thông tin liên quan trong bảng Flight\_Details sẽ bị xoá.

1. Thực hiện như sau trong QA:

|  |
| --- |
| CREATE TRIGGER Casc\_del  ON Flight  FOR DELETE  AS  DELETE Flight\_details FROM Flight\_details, DELETED  WHERE  Flight\_details.aircraft\_code=DELETED.aircraft\_code |

2. Tạo Delete Trigger khác trên bảng Flight. Trigger này sẽ thực hiện khi trigger  Casc\_del thực hiện.

|  |
| --- |
| CREATE TRIGGER del\_aircraftcode  ON Flight\_details  FOR DELETE  AS  SELECT \* FROM Flight  SELECT \* FROM Flight\_Details |

3. Thực hiện câu lệnh sau:

|  |
| --- |
| DELETE FROM Flight  WHERE aircraft\_code= ‘9W01’ |

Hình 17.6

***1.14 Tạo INSTEAD OF Trigger***

Chúng ta có thể thực hiện INSTEAD OF trigger trên bảng, nó thay thế cho câu lệnh  INSERT, UPDATE, DELETE nguyên thuỷ.

1. Thực hiện như sau bằng QA:

|  |
| --- |
| CREATE TRIGGER instead\_trigg  ON Service  INSTEAD OF INSERT  AS  BEGIN  Select Service\_code AS 'Inserted columns' From Inserted Select Service\_code AS 'Deleted columns' From Deleted Select Service\_code AS 'Table contents' From Service END |

2. Thực hiện câu lệnh sau:

|  |
| --- |
| INSERT INTO Service |

|  |
| --- |
| Values(‘FA’, ‘First Aid’) |

Kết quả:

Hình 17.7

***1.15 Bài tập***

Thực hiện những yêu cầu sau bằng QA:

1. Tạo INSERT trigger có tên ins\_chkclass trên bảng Reservation. Trigger  đảm bảo rằng dữ liệu được nhập vàp trường class code tồn tại trong bảng  Class\_master và số ghế được đặt không quá 2.

2. Tạo Cascading UPDATE trigger có tên upd\_mealcode trong bảng Meal.  Khi meal code trong bảng Meal được cập nhật thì những dữ liệu liên quan  trong bảng Airline\_meal cũng được cập nhật tương ứng.

3. Hiển thị mã lệnh (code) của trigger ins\_chkclass trên bảng Reservation.

4. Sửa trigger upd\_mealcode để người sử dụng không nhìn thấy mã lệnh của  nó.

5. Tạo View có tên all\_day chứa thông tin chi tiết sau: day code, day name,  và aircraft code trong bảng Day\_master và Flight\_days. Tạo DELETE  trigger tên là del\_day để xoá dữ liệu trong View. Kiểm tra lại sự thực hiện  của trigger.