

BÀI TẬP: Quản lý cửa hàng trà sữa

LƯỢC ĐỒ QUAN HỆ:

LOAI_MON (**MaLoai**, TenLoai)

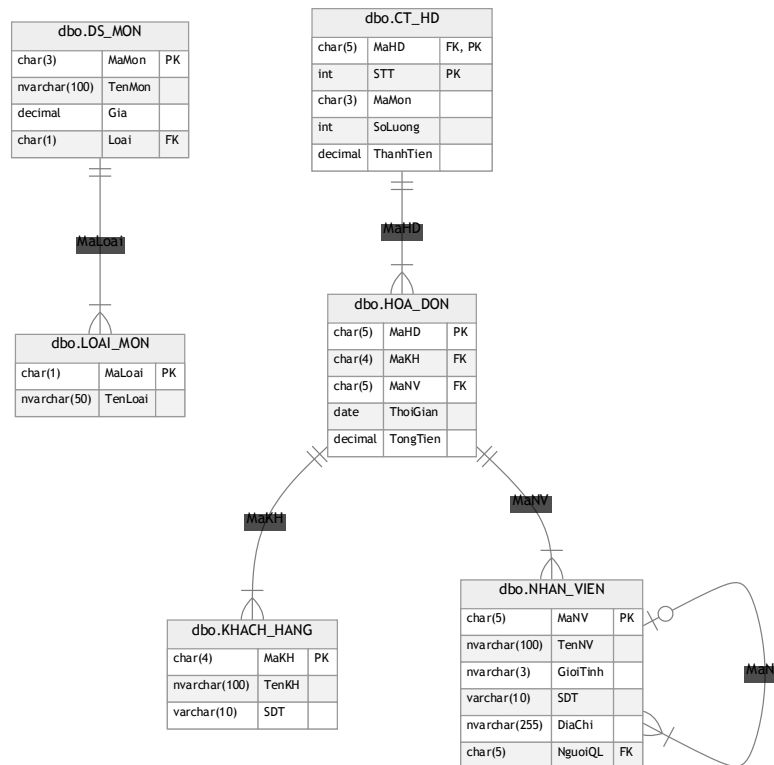
DS_MON (**MaMon**, TenMon, Gia, Loai)

KHACH_HANG (**MaKH**, TenKH, SDT)

NHAN_VIEN (**MaNV**, TenNV, GioiTinh, SDT, DiaChi, NguoiQL)

HOA_DON (**MaHD**, MaKH, MaNV, ThoiGian, TongTien)

CT_HD (**MaHD**, **STT**, MaMon, SoLuong, ThanhTien)



CƠ SỞ DỮ LIỆU

LOAI_MON

MaLoai	TenLoai
A	Cà phê
B	Trà
C	Tráng miệng

DS_MON

MaMon	TenMon	Gia	Loai
M01	Cà phê đen	20000	A
M02	Cà phê sữa	25000	A
M03	Trà đào cam sả	35000	B
M04	Matcha Latte	45000	B
M05	Bánh Donut	25000	C
M06	Bánh Flan	20000	C
M07	Trà tắc xí muội	30000	B

KHACH_HANG

MaKH	TenKH	SDT
KH01	Trần Văn Quyết	0909123123
KH02	Nguyễn Lan Anh	0909234234
KH03	Phạm Mạnh Văn	0909345345
KH04	Đoàn Hà Nhi	0909456456

NHAN_VIEN

MaNV	TenNV	GioiTinh	SDT	DiaChi	NgnoiQL
NV001	Trần Ánh Quỳnh	Nữ	0908456456	Hồng Bàng, Quận 5	NV002
NV002	Nguyễn Đông Bắc	Nam	0908234234	Trường Chinh, Tân Bình	NV004
NV003	Trần Văn Chí	Nam	0908678678	Cao Thắng, Quận 5	NV004
NV004	Phạm Thuỳ Chi	Nữ	0908123123	Nguyễn Văn Cừ, Quận 5	
NV005	Nguyễn Ánh Chi	Nữ	0908789789	Nguyễn Thị Thập, Quận 7	NV002

HOA_DON

MaHD	MaKH	MaNV	ThoiGian	TongTien
HD01	KH01	NV05	09-01-2022	115000
HD02	KH03	NV03	17-01-2022	325000
HD03	KH04	NV02	25-01-2022	185000
HD40	KH02	NV01	20-02-2022	60000

CT_HD

MaHD	STT	MaMon	SoLuong	ThanhTien
HD01	1	M01	2	40000
HD01	2	M05	3	75000
HD02	1	M02	3	75000
HD02	2	M04	4	180000
HD02	3	M03	2	70000
HD03	1	M04	1	45000
HD03	2	M02	2	50000
HD03	3	M01	1	20000
HD03	4	M03	2	70000
HD40	1	M07	2	60000

TRUY VẤN SQL

1) Câu lệnh SELECT FROM

Ví dụ 1a: Lấy danh sách tất cả các món ăn

```
SELECT * FROM DS_MON;
```

Ví dụ 1b: Chỉ lấy tên món và giá tiền

```
SELECT TenMon, Gia FROM DS_MON;
```

2) Lọc dữ liệu với WHERE

Mệnh đề WHERE lọc dữ liệu với (các) điều kiện.

- Các toán tử so sánh SQL gồm có: <, >, <=, >=, <> hoặc !=, =
- Các toán tử logic: AND OR NOT
- Toán tử BETWEEN ... AND... và IN

Ví dụ 2a: Tìm các món có giá lớn hơn 30,000

```
SELECT TenMon, Gia FROM DS_MON WHERE Gia > 30000;
```

Ví dụ 2b: Tìm ra các món ăn thuộc loại 'A' hoặc 'C'

```
SELECT *
FROM DS_MON
WHERE Loai IN ('A', 'C');
```

3) Truy vấn với điều kiện liên quan đến chuỗi ký tự, ngày tháng, Null/NotNull

a) Truy vấn với điều kiện liên quan đến chuỗi

Ví dụ 3a: Tìm các khách hàng có tên bắt đầu bằng "Nguyễn"

```
SELECT *
FROM KHACH_HANG
WHERE TenKH LIKE N'Nguyễn%';
```

Ví dụ 3b: Tìm các khách hàng tên có "Văn" ở bất kỳ vị trí nào

```
SELECT *
FROM KHACH_HANG
WHERE TenKH LIKE N'%Văn%';
```

Ví dụ 3c: Tìm các loại món có tên loại bắt đầu bằng T và có 3 ký tự

```
SELECT *
FROM LOAI_MON
WHERE TenLoai LIKE N'T__';
```

Ví dụ 3d: Tìm nhân viên có họ bắt đầu bằng "T", "H" hoặc "P"

```
SELECT *
FROM NHAN_VIEN
WHERE TenNV LIKE N'[THP]%';
```

b) Truy vấn với điều kiện liên quan đến ngày tháng

Noted: Các hàm xử lý được liệt kê dưới đây là đặc trưng của HQT MSSQL.

Hàm	Mô tả
GETDATE()	Lấy ngày giờ hiện tại của hệ thống.
DAY(d), MONTH(d), YEAR(d)	Trích xuất ngày, tháng, năm từ một giá trị ngày.
DATEDIFF(yy mm dd, d1, d2)	Tính khoảng cách giữa hai ngày theo năm, tháng hoặc ngày.
DATEPART(yy mm dd, d)	Lấy một phần cụ thể (năm, tháng, ngày) từ giá trị ngày.

MSSQL sử dụng định dạng MM/DD/YYYY, có thể thay đổi bằng SET DATEFORMAT DMY

c) Truy vấn với điều kiện liên quan đến NULL/ NOT NULL

- Sử dụng trong trường hợp:
 - Không biết (value unknown)
 - Không thể áp dụng (value inapplicable)
 - Không tồn tại (value withheld)
- Những biểu thức **tính toán** có liên quan đến giá trị NULL sẽ cho ra kết quả là **NULL**
 - x có giá trị là NULL
 - x + 3 cho ra kết quả là NULL
- Những biểu thức **so sánh** có liên quan đến giá trị NULL sẽ cho ra kết quả là **UNKNOWN**
 - x = 3 cho ra kết quả là UNKNOWN
 - x = 3 là một so sánh không hợp lệ trong SQL

4) Truy vấn trên nhiều bảng

a) Sử dụng phép tích DESCARTES

Tích Descartes trong SQL tương đương với phép nhân Cartesian Product trong toán học => tạo ra mọi cặp kết hợp có thể giữa hai tập hợp.

Ví dụ:

r	A	B
α	1	
β	2	

s	X	C	D
α	10	+	
β	10	+	
β	20	-	
γ	10	-	

r x s	A	R.B	X	C	D
α	1	α	10	+	
α	1	β	10	+	
α	1	β	20	-	
α	1	γ	10	-	
β	2	α	10	+	
β	2	β	10	+	
β	2	β	20	-	
β	2	γ	10	-	

Nếu bảng A có khóa ngoại (FK) tham chiếu đến bảng B. (Nếu bảng A không có khóa ngoại (FK) tham chiếu đến bảng B, thì khi sử dụng Tích Descartes, chúng ta sẽ kết hợp tất cả hàng của bảng A với tất cả hàng của bảng B mà không có điều kiện nào ràng buộc. => Dữ liệu bị nhân lên nhiều lần, gây tăng tải hệ thống và không có ý nghĩa thực tế nếu không có điều kiện lọc)

Ví dụ 4a: Lấy danh sách món và loại món tương ứng

```
SELECT *
FROM DS_MON, LOAI_MON
WHERE DS_MON.Loai = LOAI_MON.MaLoai;
```

b) Sử dụng điều kiện kết ở mệnh đề FROM (JOIN)

Kết bằng:

```
SELECT <danh sách các cột>
FROM R1 [INNER] JOIN R2 ON <biểu thức>
WHERE <điều kiện>
```

Ví dụ 4b: Lấy danh sách món và loại món tương ứng

```
SELECT *
FROM DS_MON
JOIN LOAI_MON ON DS_MON.Loai = LOAI_MON.MaLoai;
```

Kết ngoài

```
SELECT <danh sách các cột>
FROM R1 LEFT|RIGHT [OUTER] JOIN R2 ON <biểu thức>
```

WHERE <điều kiện>

Ví dụ 4c: Lấy tất cả khách hàng (mã và tên KH) và hoá đơn tương ứng (nếu có)

```
SELECT KHACH_HANG.MaKH, KHACH_HANG.TenKH, HOA_DON.MaHD, HOA_DON.ThoiGian
FROM KHACH_HANG
LEFT JOIN HOA_DON ON KHACH_HANG.MaKH = HOA_DON.MaKH;
```

5) Sắp xếp dữ liệu (ORDER BY)

Mệnh đề ORDER BY trong SQL được sử dụng để sắp xếp kết quả truy vấn theo một hoặc nhiều cột, theo thứ tự tăng dần (ASC) hoặc giảm dần (DESC).

Cú pháp:

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1 [ASC|DESC], column2 [ASC|DESC], ...;
```

Ví dụ 5a: Lấy thông tin khách hàng và sắp xếp tên khách hàng theo thứ tự tăng dần

```
SELECT MaKH, TenKH, SDT
FROM KHACH_HANG
ORDER BY TenKH ASC;
```

6) Phép toán tập hợp (UNION, INTERSECT, EXCEPT)

SQL có hỗ trợ các phép toán tập hợp trên 2 quan hệ **khả hợp***, gồm:

- **UNION:** Hợp hai tập dữ liệu, loại bỏ trùng lặp.
- **INTERSECT:** Giao hai tập dữ liệu (chỉ lấy các dòng xuất hiện ở cả hai).
- **EXCEPT:** Hiệu hai tập dữ liệu (lấy các dòng chỉ có trong tập đầu tiên, không có trong tập thứ hai).

Ghi chú: Quan hệ khả hợp:

- Có cùng số thuộc tính (các bảng có cùng số cột)
- Các thuộc tính tương ứng có cùng miền giá trị (các cột tương ứng phải có kiểu dữ liệu tương thích)

7) Gom nhóm (GROUP BY.. HAVING) và kết hợp (Aggregate Functions)

GROUP BY được dùng để gom nhóm dữ liệu theo một hoặc nhiều cột.

Hàm tổng hợp (Aggregate Functions) được sử dụng để tính toán trên các nhóm dữ liệu:

- COUNT() – Đếm số lượng dòng.
- SUM() – Tính tổng.
- AVG() – Tính trung bình.
- MIN(), MAX() – Tìm giá trị nhỏ nhất/lớn nhất.

HAVING được dùng để lọc kết quả sau khi nhóm, vì **WHERE không thể áp dụng trên các hàm tổng hợp**.

Ví dụ 7a: Đếm số hóa đơn của từng khách hàng

Phân tích: Nhóm các hoá đơn theo khách hàng và đếm số hoá đơn trong từng nhóm

```
SELECT KHACH_HANG.MaKH, KHACH_HANG.TenKH, COUNT(HOA_DON.MaHD) AS SoLuongHoaDon
FROM KHACH_HANG
```

```
JOIN HOA_DON ON KHACH_HANG.MaKH = HOA_DON.MaKH
```

```
GROUP BY KHACH_HANG.MaKH, KHACH_HANG.TenKH;
```

Ví dụ 7b: Lọc khách hàng có tổng chi tiêu lớn hơn 200,000

Phân tích: Gom nhóm hoá đơn theo khách hàng, tính tổng giá trị hoá đơn theo từng nhóm, lọc các nhóm có tổng lớn hơn 200,000

```
SELECT KHACH_HANG.MaKH, KHACH_HANG.TenKH, SUM(HOA_DON.TongTien) AS TongChiTieu
FROM KHACH_HANG
```

```
JOIN HOA_DON ON KHACH_HANG.MaKH = HOA_DON.MaKH
```

```
GROUP BY KHACH_HANG.MaKH, KHACH_HANG.TenKH
```

```
HAVING SUM(HOA_DON.TongTien) > 200000;
```

8) Truy vấn lồng

Cú pháp cơ bản

Câu truy vấn cha
(Outer query)

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <so sánh tập hợp> (
```

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>)
```

Câu truy vấn con
(Subquery)

Các câu lệnh SELECT có thể lồng nhau ở nhiều mức. Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic.

Mệnh đề WHERE của câu truy vấn cha có dạng: <biểu thức> <so sánh tập hợp> <truy vấn con>

Các toán tử so sánh phổ biến gồm:

- **IN, NOT IN:** kiểm tra giá trị có nằm trong tập hợp kết quả hay không.
- **ALL:** biểu thức phải thỏa điều kiện với **tất cả** giá trị trong truy vấn con.
- **ANY / SOME:** biểu thức thỏa điều kiện với ít nhất một giá trị.
- **EXISTS, NOT EXISTS:** kiểm tra sự tồn tại của dòng dữ liệu trong truy vấn con.

Có 2 loại truy vấn lồng:

- *Lồng phân cấp:*
 - Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
 - Khi thực hiện, câu truy vấn con sẽ được thực hiện trước
- *Lồng tương quan:*
 - Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
 - Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha

a) Truy vấn lồng phân cấp

Ví dụ 8a: Liệt kê các món ăn thuộc loại ‘Nước’

```
SELECT TenMon
FROM DS_MON
WHERE Loai IN (
    SELECT MaLoai
    FROM LOAI_MON
    WHERE TenLoai = N'Nước');
```

Ví dụ 8b: Tìm món ăn có giá cao hơn tất cả các món thuộc loại ‘Tráng miệng’

```
SELECT TenMon
FROM DS_MON
WHERE Gia > ALL (
    SELECT Gia
    FROM DS_MON
    WHERE Loai IN (
        SELECT MaLoai
```



```
FROM LOAI_MON
WHERE TenLoai = N'Tráng miệng'
```

```
)
```

```
);
```

b) Truy vấn lồng tương quan

Ví dụ 8c: Tìm nhân viên đã từng lập (ít nhất một) hóa đơn

```
SELECT TenNV
FROM NHAN_VIEN NV
WHERE EXISTS (
    SELECT *
    FROM HOA_DON HD
    WHERE HD.MaNV = NV.MaNV
);
```

Nhận xét IN và EXISTS:

IN:

- <tên cột> IN <câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

EXISTS

- Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS
- Nếu sub query có dữ liệu (>= 1 dòng) → tồn tại <=> mệnh đề EXISTS (...) = TRUE
- Nếu sub query không có dữ liệu (0 dòng hoặc rỗng) → Không tồn tại <=> mệnh đề EXISTS (...) = FALSE

9) Phép chia

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

S	D	E
b_i	a	1
	b	1

R÷S	A	B	C
a_i	α	a	γ
	γ	a	γ

$R \div S$ là tập các giá trị a_i trong R sao cho không có giá trị b_i nào trong S làm cho bộ (a_i, b_i) không tồn tại trong R

Hay nói cách khác, phép chia giải quyết bài toán tìm tất cả: "Tìm tất cả bộ T trong R thoả mãn tất cả S".
Để viết được câu truy vấn phép chia cần xác định 3 thành phần phép chia:

- Số bị chia: $R(X, Y)$
- Số chia: $S(Y)$
- Thương: $T(X)$

Ví dụ 9a: Tìm các khách hàng đã từng mua tất cả các món thuộc loại "Cà phê".

Phân tích: Xác định các thành phần của phép chia:

- Thương: $T(\text{MaKH})$
- Số chia: $S(\text{MaMon})$: Danh sách các món thuộc loại "Cà phê" (Kết bảng DS_MON và LOAI_MON rồi lọc các món loại "Cà phê")
- Số bị chia: $R(\text{MaKH}, \text{MaMon}) \Rightarrow$ Cần kết bảng HOA_DON và CT_HD

Có 3 cách viết câu truy vấn phép chia trong SQL, dùng GROUP BY ... HAVING, hoặc NOT EXISTS ... EXCEPT, hoặc NOT EXISTS ... NOT EXISTS ...

a) Dùng *GROUP BY ... HAVING*

SELECT R.X

FROM R

GROUP BY R.X → Trong r đếm xem ứng với mỗi bộ t có bao nhiêu bộ s thoả mãn

HAVING COUNT(R.Y) = (

SELECT COUNT(S.Y) → Đếm xem có tất cả bao nhiêu s

FROM S)

Lý giải: Ứng với mỗi nhóm $R.X_i$, nếu đếm được số $R.Y =$ tổng tất cả $Y (S.Y)$ thì bộ thứ i đó thỏa tất cả bộ trong s và là một giá trị thương cần tìm.

Ví dụ 9b: Tìm các khách hàng đã từng mua tất cả các món thuộc loại "Cà phê".

Phân tích: Nếu tổng số món thuộc loại "Cà phê" mà khách hàng thứ i đã mua = tổng số món thuộc loại cà phê trong menu, nghĩa là khách hàng đó đã mua tất cả món thuộc loại cà phê

```
SELECT KH.MaKH, KH.TenKH
FROM KHACH_HANG KH
JOIN HOA_DON HD ON KH.MaKH = HD.MaKH
JOIN CT_HD CT ON HD.MaHD = CT.MaHD
JOIN DS_MON M ON CT.MaMon = M.MaMon
JOIN LOAI_MON LM ON M.Loai = LM.MaLoai
WHERE LM.TenLoai = N'Cà phê'
GROUP BY KH.MaKH, KH.TenKH
HAVING COUNT(DISTINCT M.MaMon) = (
    SELECT COUNT(DISTINCT M2.MaMon)
    FROM DS_MON M2
    JOIN LOAI_MON LM2 ON M2.Loai = LM2.MaLoai
    WHERE LM2.TenLoai = N'Cà phê'
);
```

b) Dùng *NOT EXISTS ... EXCEPT*

```
SELECT T.X
FROM T
WHERE NOT EXISTS (SELECT S.Y -- Lấy tất cả y có
    FROM S
    EXCEPT
    SELECT R.Y -- Lấy các bộ Y thỏa T.X ở truy vấn cha
```

FROM R

WHERE R.X = T.X

Lý giải: Tìm S không thoả T:

- Nếu truy vấn con ra rỗng (not exists trả ra true): T là kết quả phép chia cần tìm ==> không có S không thoả T ==> T thoả tất cả S
- Nếu truy vấn con ra ít nhất 1 dòng (not exists trả false): T không phải kết quả phép chia cần tìm ==> có ít nhất 1 S không thoả T ==> T không thoả tất cả

Ví dụ 9c: Tìm các khách hàng đã từng mua tất cả các món thuộc loại "Cà phê".

SELECT KH.MaKH, KH.TenKH

FROM KHACH_HANG KH

WHERE NOT EXISTS (

SELECT M.MaMon

FROM DS_MON M

JOIN LOAI_MON LM ON M.Loai = LM.MaLoai

WHERE LM.TenLoai = N'Cà phê'

EXCEPT

SELECT CT.MaMon

FROM HOA_DON HD

JOIN CT_HD CT ON HD.MaHD = CT.MaHD

WHERE HD.MaKH = KH.MaKH

)

AND EXISTS (-- *Kiểm tra tập S khác rỗng, nếu S rỗng trả về tất cả R(X)*

SELECT 1

FROM DS_MON M

JOIN LOAI_MON LM ON M.Loai = LM.MaLoai

```
WHERE LM.TenLoai = N'Cà phê'
```

```
);
```

c) Dùng *NOT EXISTS ... NOT EXISTS ...*

```
SELECT T.X
```

```
FROM T
```

```
WHERE NOT EXISTS ( -- Tìm xem có S.Y nào không thoã không
```

```
    SELECT *
```

```
    FROM S
```

```
    WHERE NOT EXISTS (
```

```
        SELECT *
```

```
        FROM R
```

```
        WHERE R.X = T.X AND R.Y = S.Y
```

```
    ) -- Nếu không tồn tại => S.Y không thoã
```

```
);
```

Lý giải: Với mỗi T.X kiểm tra xem có S.Y nào không thoã S hay không

Ví dụ 9d: Tìm các khách hàng đã từng mua tất cả các món thuộc loại "Cà phê".

Phân tích: Nếu khách hàng i mua tất cả các món Cà phê, nghĩa là không tồn tại món cà phê nào mà khách hàng đó chưa từng mua

```
SELECT KH.MaKH, KH.TenKH
```

```
FROM KHACH_HANG KH
```

```
WHERE NOT EXISTS (
```

```
    SELECT M.MaMon
```

```
    FROM DS_MON M
```

```
    JOIN LOAI_MON LM ON M.Loai = LM.MaLoai
```

```
    WHERE LM.TenLoai = N'Cà phê'
```

```
    AND NOT EXISTS (
```

```
        SELECT *
```

```

FROM HOA_DON HD
JOIN CT_HD CT ON HD.MaHD = CT.MaHD
WHERE HD.MaKH = KH.MaKH AND CT.MaMon = M.MaMon
)
)
AND EXISTS ( -- Kiểm tra tập S khác rỗng, nếu S rỗng trả về tất cả R(X)
SELECT 1
FROM DS_MON M
JOIN LOAI_MON LM ON M.Loai = LM.MaLoai
WHERE LM.TenLoai = N'Cà phê'
);

```

10) Cập nhật dữ liệu

a) INSERT

Dùng để thêm 1 hay nhiều dòng vào bảng.

Cú pháp (thêm 1 dòng):

INSERT INTO <tên bảng>(<danh sách các thuộc tính>)

VALUES (<danh sách các giá trị>)

Nhận xét:

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
- Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV:
 - Khóa chính
 - Tham chiếu
 - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

Cú pháp (thêm nhiều dòng):

INSERT INTO <tên bảng>(<danh sách các thuộc tính>)

<câu truy vấn con>

b) DELETE

Dùng để xóa các dòng của bảng.

DELETE FROM <tên bảng>

[WHERE <điều kiện>]

Nhận xét:

- Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
- Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
 - Không cho xóa
 - Xóa luôn những dòng có giá trị đang tham chiếu đến: CASCADE
 - Đặt NULL cho những giá trị tham chiếu

c) UPDATE

Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng.

UPDATE <tên bảng>

SET <tên thuộc tính>=<giá trị mới>,

<tên thuộc tính>=<giá trị mới>,

...

[WHERE <điều kiện>]

Nhận xét:

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
- Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
 - Không cho sửa
 - Sửa luôn những dòng có giá trị đang tham chiếu đến
 - CASCADE

GHI CHÚ:

- Sinh viên đọc thêm về Khung nhìn và Chỉ mục trong slides

Bài tập luyện tập

1. Lấy danh sách tất cả họ tên nhân viên cùng với mã số quản lý của họ.
2. Truy vấn danh sách các hóa đơn cùng với tổng tiền của từng hóa đơn.
3. Lấy danh sách nhân viên có giới tính là Nữ.
4. Lấy danh sách các món có giá trong khoảng từ 20,000 VND đến 40,000 VND.
5. Lấy danh sách các món có mã là M01, M02, M06 hoặc M08.

6. Tìm món có chữ "Trà" ở bất kỳ vị trí nào.
7. Tìm khách hàng có họ gồm đúng 3 ký tự và bắt đầu bằng "T".
8. Tìm nhân viên có địa chỉ ở Quận 5.
9. Lấy danh sách hóa đơn với thông tin khách hàng (tên) tương ứng.
10. Lấy danh sách món ăn và loại món của chúng.
11. Lấy danh sách tất cả nhân viên và người quản lý của họ (nếu có).
12. Lấy danh sách hóa đơn có tổng tiền lớn hơn 100,000 VND và được tạo bởi nhân viên "Nguyễn Đông Bắc".
13. Lấy danh sách khách hàng có số điện thoại bắt đầu bằng "0909" hoặc chưa có hóa đơn.
14. Lấy danh sách nhân viên đã tạo hóa đơn trong tháng 1 hoặc có tổng tiền hóa đơn trên 200,000 VND.
15. Lấy danh sách món ăn có giá từ 20,000 đến 40,000 VND và thuộc loại "Trà".
16. Lấy danh sách nhân viên có hóa đơn trong tháng 1 hoặc tháng 2, tổng tiền hóa đơn trên 150,000 VND, và có địa chỉ chứa từ "Quận"
17. Lấy danh sách tất cả khách hàng (mã và họ tên khách hàng) đã từng đặt trà hoặc cà phê bất kỳ.
18. Cho biết mã hoá đơn do khách hàng có họ là "Nguyễn" đặt hoặc nhân viên lập hoá đơn có họ là "Nguyễn", sắp xếp tăng dần theo mã hoá đơn.
19. Lấy danh sách mã khách hàng đã mua hàng nhưng chưa từng đặt món nào thuộc loại "Tráng miệng".
20. Lấy danh sách các món đã từng được đặt ít nhất một lần và thuộc loại "Trà".
21. Lấy danh sách hóa đơn có tổng tiền trên 100,000 VND và chứa ít nhất một món "Cà phê sữa".
22. Tính tổng tiền từng khách hàng đã chi tiêu.
23. Tính số món đã bán và giá trung bình của từng loại món.
24. Tìm món ăn (mã món, tên món) đã xuất hiện trong hơn 2 hóa đơn.
25. Lọc nhân viên (mã, họ tên) có tổng doanh số trên 300,000.
26. Tìm hóa đơn có tổng tiền cao hơn 100,000 và chứa ít nhất 2 món.
27. Liệt kê tên các món ăn có giá lớn hơn tất cả các món trong loại "Tráng miệng".
28. Tìm các món ăn có giá thấp hơn ít nhất một món trong loại "Cà phê".
29. Tìm tên khách hàng đã từng mua món "Bánh Donut".
30. Liệt kê các nhân viên chưa từng lập hóa đơn nào chứa món có giá > 200,000.
31. Tìm tên những món ăn có đơn giá cao hơn *mọi món* mà khách hàng "Nguyễn Văn A" đã từng mua.
32. Tìm những nhân viên quản lý 2 nhân viên trở lên và có lập hoá đơn trên 200,000.
33. Tìm các món (mã, tên món) có trong tất cả các hoá đơn.
34. Tìm khách hàng mua tất cả các món thuộc loại trà.
35. Tìm khách hàng đã mua tất cả các món thuộc loại "Nước".
36. Tìm nhân viên đã lập hóa đơn cho tất cả các khách hàng có trong hệ thống.
37. Tìm khách hàng đã từng mua *ít nhất một món* ở mỗi loại món ăn có trong nhà hàng.
38. Tìm khách hàng đã mua *mọi món* mà khách hàng "Nguyễn Văn A" đã từng mua.

39. Tìm nhân viên đã lập hóa đơn có chứa tất cả các món thuộc loại "Món chính", *trong cùng một hóa đơn bất kỳ*.