**Hệ thống thông tin quản lý của hàng Circle K**

Dưới đây là hệ thống thông tin chi tiết quản lý các hoạt động và thông tin của cửa hàng Circle K. Hệ thống này bao gồm các bảng lưu trữ thông tin về nhà cung cấp, sản phẩm, nhóm hàng, nhiên viên, khách hàng, hóa đơn, khuyến mãi, phản hồi và doanh thu.

**Xác định các thự thể và thuộc tính**

**1. Nhà cung cấp sản phẩm (SUPPLIER)**

* + MaNCC (INT, PK): Mã nhà cung cấp (duy nhất).
  + TenNCC (VARCHAR(100)): Tên nhà cung cấp.
  + DiaChi (VARCHAR(255)): Địa chỉ của nhà cung cấp.
  + SDT (VARCHAR(15)): Số điện thoại liên hệ.
  + Email (VARCHAR(100)): Email liên hệ.

**2. Sản phẩm (PRODUCT)**

* + MaSP (INT, PK): Mã sản phẩm (duy nhất).
  + TenSP (VARCHAR(100)): Tên sản phẩm.
  + DonGia (DECIMAL(10, 2)): Đơn giá sản phẩm.
  + HanSuDung (DATE): Hạn sử dụng của sản phẩm.
  + NoiSanXuat (VARCHAR(100)): Nơi sản xuất.
  + MaNhom (INT, FK): Mã nhóm hàng, tham chiếu đến bảng CATEGORY.

**3. Nhóm hàng (CATAGORY)**

* + MaNhom (INT, PK): Mã nhóm hàng (duy nhất).
  + TenNhom (VARCHAR(100)): Tên nhóm hàng.

**4. Nhân viên (EMPLOYEE)**

* + MaNV (INT, PK): Mã nhân viên (duy nhất).
  + HoTen (VARCHAR(100)): Họ và tên nhân viên.
  + GioiTinh (ENUM): Giới tính (Nam, Nữ).
  + NgaySinh (DATE): Ngày sinh của nhân viên.
  + DiaChi (VARCHAR(255)): Địa chỉ của nhân viên.
  + SDT (VARCHAR(15)): Số điện thoại liên hệ.
  + MaCV (INT, FK): Mã chức vụ, tham chiếu đến bảng JOB\_TITLE.

**5. Chức vụ (JOB\_TITLE)**

* + MaCV (INT, PK): Mã chức vụ (duy nhất).
  + TenCV (VARCHAR(100)): Tên chức vụ.

**6. Quyền đăng nhập (USER\_ACCOUNT)**

* + MaNV (INT, PK, FK): Mã nhân viên, tham chiếu đến bảng EMPLOYEE.
  + TenDangNhap (VARCHAR(50)): Tên đăng nhập.
  + MatKhau (VARCHAR(50)): Mật khẩu.

**7. Khách hàng (CUSTOMER)**

* + MaKH (INT, PK): Mã khách hàng (duy nhất).
  + HoTen (VARCHAR(100)): Họ và tên khách hàng.
  + SDT (VARCHAR(15)): Số điện thoại.
  + SoLanMua (INT): Số lần mua hàng.

**8. Phản hồi khách hàng (FEEDBACK)**

* + MaPH (INT, PK): Mã phản hồi (duy nhất).
  + MaKH (INT, FK): Mã khách hàng, tham chiếu đến bảng CUSTOMER.
  + NgayPhanHoi (DATE): Ngày khách hàng gửi phản hồi.
  + NoiDung (VARCHAR(255)): Nội dung phản hồi.
  + SoDiemDanhGia (INT): Số điểm đánh giá từ khách hàng (1-5).

**9. Hóa đơn (INVOICE)**

* + MaHD (INT, PK): Mã hóa đơn (duy nhất).
  + MaKH (INT, FK): Mã khách hàng, tham chiếu đến bảng CUSTOMER.
  + NgayLap (DATE): Ngày lập hóa đơn.
  + TongSoTien (DECIMAL(10, 2)): Tổng số tiền trong hóa đơn.

**10. Khuyến mãi (PROMOTION)**

* + MaKM (INT, PK): Mã khuyến mãi (duy nhất).
  + NoiDung (VARCHAR(255)): Nội dung khuyến mãi.
  + ThoiGianBatDau (DATE): Thời gian bắt đầu khuyến mãi.
  + ThoiGianKetThuc (DATE): Thời gian kết thúc khuyến mãi.
  + DoiTuongApDung (VARCHAR(100)): Đối tượng áp dụng khuyến mãi.
  + MucGiamGia (DECIMAL(5, 2)): Mức giảm giá (%).

**Mối liên kết giữa các thực thể**

1. SUPPLIER và PRODUCT: quan hệ 1-n vì một nhà cung cấp có thể cung cấp nhiều sản phẩm và một sản phẩn chỉ có một nhà cung cấp trong cửa hàng Circle K.
2. CATAGORY và PRODUCT: quan hệ 1-n vì một nhóm hàng có thể chứa nhiều sản phẩm và một sản phẩm chỉ thuộc một nhóm hàng.
3. EMPLOYEE và JOB\_TITLE: quan hệ n-1 vì một nhần viên chỉ thuộc một chức vụ trong của hàng và một chức vụ trong cửa hàng có thể có nhiều nhân viên.
4. EMPLOYEE và USER\_ACCOUNT: quan hệ 1-1 vì mỗi nhân viên có một tài khoản đăng nhập duy nhất.
5. CUSTOMER và INVOICE: quan hệ 1-n vì một khách hàng có thể có nhiều hóa đơn và một hóa đơn thì chỉ thuộc về một khách hàng.
6. CUSTOMER và FEEDBACK: quan hệ 1-n vì một khách hàng có thể gửi nhiều phản hồi và một phản hồi chỉ do một khách hàng phản hồi.
7. PROMOTION và PRODUCT: quan hệ 1-n vì mội khuyến mãi có thể áp dụng cho nhiều sản phẩm và một sản phẩm thì chỉ có thể có một mã khuyến mãi.

**Chuẩn hóa và quan hệ ứng viên**

Hệ thống thông tin này tuân theo chuẩn 3NF:

1. Mỗi thực thể đều có **khóa chính** duy nhất
2. Các thuộc tính không khó đều phụ thuộc và khóa chính, không phụ thuộc bắc cầu.
3. Không có thuộc tính đa trị hoặc nhóm thuộc tính lặp.

**Lệnh SQL tạo cơ sở dữ liệu trên MS SQL Server 2022**

|  |
| --- |
| -- tạo cơ sở dữ liệu  CREATE DATABASE StoreCircleK;  GO  -- Sử dụng cơ sở dữ liệu  USE StoreCircleK;  GO  -- tạo bảng Supplier  CREATE TABLE SUPPLIER (  MaNCC INT PRIMARY KEY IDENTITY(1,1),  TenNCC VARCHAR(100) NOT NULL,  DiaChi VARCHAR(255),  SDT VARCHAR(15),  Email VARCHAR(100)  );  Go  -- tạo bảng Category  CREATE TABLE CATEGORY (  MaNhom INT PRIMARY KEY IDENTITY(1,1),  TenNhom VARCHAR(100) NOT NULL  );  Go  -- tạo bảng Product  CREATE TABLE PRODUCT (  MaSP INT PRIMARY KEY IDENTITY(1,1),  TenSP VARCHAR(100) NOT NULL,  DonGia DECIMAL(10, 2) NOT NULL,  HanSuDung DATE,  NoiSanXuat VARCHAR(100),  MaNhom INT,  MaNCC INT,  FOREIGN KEY (MaNhom) REFERENCES CATEGORY(MaNhom),  FOREIGN KEY (MaNCC) REFERENCES SUPPLIER(MaNCC)  );  Go  -- tạo bảng Job\_Title  CREATE TABLE JOB\_TITLE (  MaCV INT PRIMARY KEY IDENTITY(1,1),  TenCV VARCHAR(100) NOT NULL  );  Go  CREATE TABLE EMPLOYEE (  MaNV INT PRIMARY KEY IDENTITY(1,1),  HoTen VARCHAR(100) NOT NULL,  GioiTinh CHAR(3) CHECK (GioiTinh IN ('Nam', 'Nu')) NOT NULL,  NgaySinh DATE,  DiaChi VARCHAR(255),  SDT VARCHAR(15),  MaCV INT,  FOREIGN KEY (MaCV) REFERENCES JOB\_TITLE(MaCV)  );  Go  -- tạo bảng Account  CREATE TABLE USER\_ACCOUNT (  MaNV INT PRIMARY KEY,  TenDangNhap VARCHAR(50) NOT NULL,  MatKhau VARCHAR(50) NOT NULL,  FOREIGN KEY (MaNV) REFERENCES EMPLOYEE(MaNV)  );  Go  -- tạo bảng Customer  CREATE TABLE CUSTOMER (  MaKH INT PRIMARY KEY IDENTITY(1,1),  HoTen VARCHAR(100) NOT NULL,  SDT VARCHAR(15) NOT NULL,  SoLanMua INT DEFAULT 0  );  Go  -- tạo bảng FeedBack  CREATE TABLE FEEDBACK (  MaPH INT PRIMARY KEY IDENTITY(1,1),  MaKH INT,  NgayPhanHoi DATE NOT NULL,  NoiDung VARCHAR(255),  SoDiemDanhGia INT CHECK (SoDiemDanhGia BETWEEN 1 AND 5),  FOREIGN KEY (MaKH) REFERENCES CUSTOMER(MaKH)  );  Go  -- tạo bảng Invoice  CREATE TABLE INVOICE (  MaHD INT PRIMARY KEY IDENTITY(1,1),  MaKH INT,  NgayLap DATE NOT NULL,  TongSoTien DECIMAL(10, 2) NOT NULL,  FOREIGN KEY (MaKH) REFERENCES CUSTOMER(MaKH)  );  Go  -- tạo bảng Promotion  CREATE TABLE PROMOTION (  MaKM INT PRIMARY KEY IDENTITY(1,1),  NoiDung VARCHAR(255) NOT NULL,  ThoiGianBatDau DATE,  ThoiGianKetThuc DATE,  DoiTuongApDung VARCHAR(100),  MucGiamGia DECIMAL(5, 2) NOT NULL  ); |

**Tóm tắt**

Hệ thống quản lý này bao gồm việc quản lý thông tin về:

* **Nhà cung cấp, sản phẩm, nhóm hàng** giúp quản lý nguồn cung và phân loại sản phẩm.
* **Nhân viên, chức vụ, quyền đăng nhập** giúp quản lý quyền hạn và thông tin nhân viên.
* **Khách hàng, phản hồi, hóa đơn** giúp theo dõi thông tin khách hàng và lịch sử mua săm.
* **Khuyến mãi** giúp quản lý các chương trình khuyến mãi áp dụng cho sản phẩm.

Hệ thống thông tin quản lý của hàng Cirlce K này đảm bảo quản lý chi tiết và dễ dàng mở rộng khi có thêm yêu cầu mới. Dưới đây là một số lệnh để thêm dữ liệu ban đầu vào các dòng trong hệ thống thông tin quản lý Circle K. Những lệnh này sẽ giúp khởi tạo hệ thống với dữ liệu mẫu để dễ dàng kiểm tra và chạy thử.

|  |
| --- |
| -- thêm dữ liệu vào bảng Supplier  INSERT INTO SUPPLIER (TenNCC, DiaChi, SDT, Email) VALUES  ('Vinamilk', '10 Tan Trao, Phuong Tan Pha, Quan 7, Tp.HCM', '02854155555', 'vinamilk@gmail.com'),  ('Coca-Cola', 'Xa lo Ha Noi, Phuong Linh Trung, Tp.Thu Đuc', '1900555584', 'cocacola@gmail.com'),  ('PepsCo', N'88 Đuong Le Van Viet, Phuong Hiep Phu, Quan 9, Tp.HCM', '02862890000', 'pepsico@gmail.com'),  ('Acecook Viet Nam', 'Lo II-3, Đuong số II, Khu Cong Nghiep Tan Binh, Quan Tan Phu, Tp.HCM', NULL, 'acecook@gmail.com');  -- thêm dữ liệu vào bảng Category  INSERT INTO CATEGORY (TenNhom)  VALUES  ('Đo uong'),  ('Đo an');  -- thêm dữ liệu vào bảng Product  INSERT INTO PRODUCT (TenSP, DonGia, HanSuDung, NoiSanXuat, MaNhom, MaNCC)  VALUES  ('Sua tuoi 100% Vinamilk', 30000, '2025-04-12', 'Trang trai bo sua tai Viet Nam', 1, 1),  ('Yomilk', 20000, '2025-03-13', 'Trang trai bo sua tai Viet Nam', 1, 1),  ('Fanst', 12000, '2024-12-04', 'Nha may Coca - Cola tai Viet Nam', 1, 2),  ('7UP', 10000, '2025-01-05', 'Nha may của Suntory PepsiCo Viet Nam', 1, 3),  ('Pho Nhip Song', 9000, '2025-01-22', 'Nha may cua Acecook Viet Nam', 2, 4),  ('My Hao Hao', 4000, '2025-03-19', 'Nha may cua Acecook Viet Nam', 2, 4);  -- thêm dữ liệu vào bảng Job\_Title  INSERT INTO JOB\_TITLE (TenCV)  VALUES  ('Nhan vien ban hang'),  ('Quan ly cua hang'),  ('Nhan Vien Don Dep');  -- thêm dữ liệu vào bản Employee  INSERT INTO EMPLOYEE (HoTen, GioiTinh, NgaySinh, DiaChi, SDT, MaCV)  VALUES  ('Nguyen Vu Triet ', 'Nam', '2005-05-02', 'Bu To, Phuoc tan, Phu Rieng, Binh Phuoc', '0367182578', 1),  ('Bui Thi Thu Thao', 'Nu', '2005-07-11', 'Thon 1, Đuc Lieu, Bu Đang, Binh Phuoc', '0934567890', 2),  ('Truong Thanh Long', 'Nam', '2005-03-28', 'Thon 1, Đuc Lieu, Bu Đang, Binh Phuoc', '0922334455', 1),  ('Nguyen Thi Phuong Anh', 'Nu', '2005-05-19', 'Thôn 2, Nghia Trung, Bu Đang, Binh Phuoc', '0933334651', 3);  -- thêm dữ liệu vào bảng User\_Account  INSERT INTO USER\_ACCOUNT (MaNV, TenDangNhap, MatKhau)  VALUES  (1, 'nguyenvutriet', 'pw01'),  (2, 'thuthao', 'pw02'),  (3, 'thanhlong', 'pw03'),  (4, 'phuonganh', 'pw04');  -- thêm dữ liệu vào bảng Customer  INSERT INTO CUSTOMER ( HoTen, SDT, SoLanMua)  VALUES  ('Mai Thi Hue', '0909998887', 10),  ( 'Phan Tran Huyen Trang', '0912349876', 2),  ( 'Pham Ngoc Nam', '0923456789', 5),  ('Le Minh Quan', '0998772116', 1);  -- thêm dữ liệu vào bảng Feedback  INSERT INTO FEEDBACK (MaKH, NgayPhanHoi, NoiDung, SoDiemDanhGia)  VALUES  ( 1 , '2024-12-01', 'San pham rat ngon, se tiep tuc ung ho!', 5),  ( 3, '2024-12-05', 'Yomilk rat ngon, sẽ ung ho!', 4),  ( 4, '2024-12-25', 'Cua hang trang tri đep!', 4);  -- thêm dữ liệu vào bảng Invoice  INSERT INTO INVOICE ( MaKH, NgayLap, TongSoTien)  VALUES  ( 1, '2024-11-19', 234000),  ( 2, '2024-11-20', 25000),  ( 3, '2024-11-25', 36000),  ( 4, '2024-11-28', 51000),  (2,'2024-11-21', 20000),  (4,'2024-11-26', 13000);  -- thêm dữ liệu vào bản Promotion  INSERT INTO PROMOTION (NoiDung, ThoiGianBatDau, ThoiGianKetThuc, DoiTuongApDung, MucGiamGia)  VALUES  ('Giam gia 10% cho cac looi đo uong', '2024-11-01', '2024-12-01', 'Đo uong', 10.00),  ('Mua 5 goi my hao hao đuoc tang ve xem concert Adele o Munich', '2024-11-18', '2024-11-19', null, 0.00),  ('Giam 5% cho cac loai đo an', '2024-11-18', '2024-11-25', 'Đo an', 5.00); |

**Tóm tắt**

Các lệnh trên đã giúp tạo dữ liệu ban đầu cho hệ thống quản lý cửa hàng bán đồ ăn vặt:

* **Nhà cung cấp**, **nhóm hàng**, **sản phẩm** giúp quản lý nguồn cung và sản phẩm.
* **Nhân viên**, **chức vụ**, **quyền đăng nhập** giúp quản lý thông tin và phân quyền cho nhân viên.
* **Khách hàng**, **hóa đơn**, **phản hồi** giúp theo dõi lịch sử mua hàng và ý kiến từ khách hàng.
* **Khuyến mãi** giúp quản lý các chương trình khuyến mãi áp dụng cho sản phẩm.

Với dữ liệu ban đầu này, hệ thống đã sẵn sàng để sử dụng và kiểm tra các chức năng cơ bản.

Dưới đây là **20 câu truy vấn SQL** cho hệ thống thông tin cửa hàng Circle K, được sắp xếp từ dễ đến khó. Mỗi truy vấn tập trung vào các truy vấn hoàn toàn khác nhau, giúp ta hiểu rõ về các truy vấn phổ biến.

**1. Thông tin nhân viên của Circle K**

|  |
| --- |
| SELECT HoTen, GioiTinh, NgaySinh, DiaChi FROM EMPLOYEEs; |

**2. Tìm nhân viên bán hàng**

|  |
| --- |
| SELECT HoTen FROM EMPLOYEEs WHERE MaCV = 1; |

**3. Tên và số điện thoại của khách hàng có số lượt mua trên 5**

|  |
| --- |
| SELECT HoTen, SDT FROM CUSTOMERs WHERE SoLanMua > 5; |

**4. Lời phản hồi của khách hàng có mã khách hàng là 3 và lượt đánh giá từ trên 4**

|  |
| --- |
| SELECT NgayPhanHoi, NoiDung FROM FEEDBACKs WHERE MaKH = 3 AND SoDiemDanhGia >= 4; |

**5. Tên của khách hàng đã đánh giá 5 sao**

|  |
| --- |
| SELECT c.HoTen, c.SDT  FROM CUSTOMERs c  JOIN FEEDBACKs f ON c.MaKH = f.MaKH  WHERE f.SoDiemDanhGia = 5; |

**6. Hóa đơn kèm theo tên của người mua**

|  |
| --- |
| SELECT c.HoTen, i.TongSoTien  FROM INVOICEs i  JOIN CUSTOMERs c ON i.MaKH = c.MaKH; |

**7. Những sản phẩm của Vinamilk có đơn giá trên 25000**

|  |
| --- |
| SELECT p.TenSP, p.DonGia  FROM PRODUCTs p  JOIN SUPPLIERs s ON p.MaNCC = s.MaNCC  WHERE s.TenNCC = 'Vinamilk' AND p.DonGia > 25000; |

**8. Đếm số lượng nhân viên trong Circle K**

|  |
| --- |
| SELECT COUNT(\*) FROM EMPLOYEEs; |

**9. Tổng số sản phẩm của mỗi nhà cung cấp**

|  |
| --- |
| SELECT s.TenNCC, COUNT(p.MaSP) AS TongSanPham  FROM SUPPLIERs s  JOIN PRODUCTs p ON s.MaNCC = p.MaNCC  GROUP BY s.TenNCC; |

**10. Tổng số tiền trung bình của các khách hàng**

|  |
| --- |
| SELECT AVG(TongSoTien)  FROM INVOICEs; |

**11. Phiếu khuyến mãi được áp dụng trong thời gian hiện tại**

|  |
| --- |
| SELECT NoiDung, MucGiamGia, ThoiGianBatDau, ThoiGianKetThuc  FROM PROMOTIONs  WHERE ThoiGianBatDau <= GETDATE() AND ThoiGianKetThuc >= GETDATE(); |

**12. Sản phẩm hết hạn trong 30 ngày tính từ ngày hiện tại**

|  |
| --- |
| SELECT TenSP, HanSuDung, NoiSanXuat  FROM PRODUCTs  WHERE HanSuDung <= DATEADD(DAY, 30, GETDATE()); |

**13. Mỗi tên công việc lấy ra tên nhân viên trong đó**

|  |
| --- |
| SELECT j.TenCV, e.HoTen  FROM JOB\_TITLEs j  JOIN EMPLOYEEs e ON e.MaCV = j.MaCV; |

**14. Tính tổng đơn giá của các sản phẩm của nhà cung cấp**

|  |
| --- |
| SELECT s.TenNCC, SUM(p.DonGia) AS TongDonGia  FROM SUPPLIERs s  JOIN PRODUCTs p ON s.MaNCC = p.MaNCC  GROUP BY s.TenNCC; |

**15. Lấy ra sản phẩm thuộc nhóm đồ uống và sắp xếp theo đơn giá giảm dần**

|  |
| --- |
| SELECT p.TenSP, p.DonGia  FROM PRODUCTs p  JOIN CATEGORies c ON p.MaNhom = c.MaNhom  WHERE c.TenNhom = 'Đo uong'  ORDER BY p.DonGia DESC; |

**16. Lấy ra sản phẩm thuộc nhóm đồ ăn và sắp xếp theo đơn giá tăng dần**

|  |
| --- |
| SELECT p.TenSP, p.DonGia  FROM PRODUCTs p  JOIN CATEGORies c ON p.MaNhom = c.MaNhom  WHERE c.TenNhom = 'Đo an'  ORDER BY p.DonGia ASC; |

**17. Thông tin khách hàng đã đánh giá và sắp xếp theo số lần mua**

|  |
| --- |
| SELECT c.HoTen, c.SoLanMua, f.NoiDung  FROM CUSTOMERs c  JOIN FEEDBACKs f ON c.MaKH = f.MaKH  WHERE f.SoDiemDanhGia IS NOT NULL  ORDER BY c.SoLanMua; |

18. Thông tin khách hàng không đánh giá của hàng Circle K

|  |
| --- |
| SELECT c.HoTen, c.SDT  FROM CUSTOMERs c  LEFT JOIN FEEDBACKs f ON c.MaKH = f.MaKH  WHERE f.SoDiemDanhGia IS NULL; |

**19. Thông tin khách hàng mua đơn dưới 100k kèm theo lời phản hồi**

|  |
| --- |
| SELECT c.HoTen, f.NoiDung, SUM(i.TongSoTien) AS TongTien  FROM INVOICEs i  JOIN CUSTOMERs c ON i.MaKH = c.MaKH  LEFT JOIN FEEDBACKs f ON c.MaKH = f.MaKH  GROUP BY c.HoTen, f.NoiDung  HAVING SUM(i.TongSoTien) < 100000; |

**20. Tổng số tiền của các hóa đơn theo khách hàng và sắp xếp theo thứ tự giảm dần**

|  |
| --- |
| SELECT c.HoTen, c.SDT, SUM(i.TongSoTien) AS TongTien  FROM INVOICEs i  JOIN CUSTOMERs c ON i.MaKH = c.MaKH  GROUP BY c.HoTen, c.SDT  ORDER BY TongTien DESC; |

**Giải thích một số câu lệnh:**

* **Sử dụng Join để liên kết các bảng**: Các câu truy vấn như 5, 6, 7, 9, 13, 14, 15, 16, 17, 18, 19 và 20 yêu cầu hiểu về Join để nối dữ liệu từ nhiều bảng.
* **Tính toán trên dữ liệu**: Các câu lệnh như 8, 14, 19, và 20 yêu cầu sử dụng các hàm tổng hợp (Count, Sum).
* **Lọc dữ liệu với điều kiện**: Sử dụng các điều kiện where kết hợp với các hàm AND, NOW, CURDATE() + INTERVAL 30 DAY và IS NOT NULL để lọc dữ liệu theo yêu cầu của đề bài.

Những truy vấn này giúp ta làm quen với các dạng câu hỏi SQL phổ biến, từ các truy vấn đơn giản đến các truy vấn phức tạp.

Dưới đây là các truy vấn SQL trên được chuyển thành các biểu thức đại số quan hệ, giúp hiểu rõ cách viết và biểu diễn bằng đại số quan hệ.

**1. Thông tin nhân viên của Circle K.**

πHoTen, GioiTinh, NgaySinh, DiaChi​(EMPLOYEEs)

**2. Tìm nhân viên bán hàng.**

πHoTen​(σMaCV=1​(EMPLOYEEs))

**3. Tên và số điện thoại của khách hàng có số lượt mua trên 5.**

πHoTen, SDT​(σSoLanMua>5​(CUSTOMERs))

**4. Lời phản hồi của khách hàng có mã khách hàng là 3 và lượt đánh giá từ 4 sao.**

πNgayPhanHoi, NoiDung​(σMaKH=3∧SoDiemDanhGia≥4​(FEEDBACKs))

**5. Tên của khách hàng đã đánh giá 5 sao.**

πHoTen, SDT​(σSoDiemDanhGia=5​(CUSTOMERs⋈CUSTOMERs.MaKH=FEEDBACKs.MaKH​FEEDBACKs))

**6. Hóa đơn kèm theo tên của người mua.**

πHoTen, TongSoTien​(INVOICEs⋈INVOICEs.MaKH=CUSTOMERs.MaKH​CUSTOMERs)

**7. Những sản phẩm của Vinamilk có đơn giá trên 25000.**

πTenSP, DonGia​(σTenNCC=′Vinamilk′∧DonGia>25000​(PRODUCTs⋈PRODUCTs.MaNCC=SUPPLIERs.MaNCC​SUPPLIERs))

**8. Đếm số lượng nhân viên trong Circle K.**

*Đại số quan hệ không hỗ trợ việc đếm nhân viên trong Circle K.*

**9. Tổng số sản phẩm của mỗi nhà cung cấp.**

*Đại số quan hệ không hỗ trợ việc đếm số sản phẩm của nhà cung cấp.*

**10. Tổng số tiền trung bình của các khách hàng.**

*Đại số quan hệ không hỗ trợ tính tổng và đếm.*

**11. Phiếu khuyến mãi được áp dụng trong thời gian hiện tại.**

πNoiDung, MucGiamGia, ThoiGianBatDau, ThoiGianKetThuc​(σThoiGianBatDau≤NOW∧ThoiGianKetThuc≥NOW​(PROMOTIONs))

**12. Sản phẩm hết hạn trong 30 ngày tính từ ngày hiện tại.**

πTenSP, HanSuDung, NoiSanXuat​(σHanSuDung≤CURDATE+30​(PRODUCTs))

**13. Mỗi tên công việc lấy ra tên nhân viên trong đó.**

πTenCV,HoTen​(σe.MaCV=j.MaCV​(JOB\_TITLEs×EMPLOYEEs))

**14. Tính tổng đơn giá của các sản phẩm của nhà cung cấp.**

*Đại số quan hệ không hỗ trợ tính tổng nên không thể tính tổng đơn giá được.*

**15. Lấy ra sản phẩm thuộc nhóm đồ uống và sắp xếp theo đơn giá giảm dần.**

*Đại số quan hệ không hỗ trợ việc sắp xếp giảm dần.*

**16. Lấy ra sản phẩm thuộc nhóm đồ ăn và sắp xếp theo đơn giá tăng dần.**

*Đại số quan hệ trong hỗ trợ việc sắp xếp tăng dần hay giảm dần.*

**17. Thông tin khách hàng đã đánh giá và sắp xếp theo số lần mua.**

*Đại số quan hệ không hỗ trợ việc sắp xếp.*

**18. Thông tin khách hàng không đánh giá của hàng Circle K.**

πHoTen, SDT​(σSoDiemDanhGia=NULL​(CUSTOMERs LEFT OUTER JOIN FEEDBACKs))

**19. Thông tin khách hàng mua đơn dưới 100k kèm theo lời phản hồi.**

*Đại số quan hệ không hỗ trợ phép tính tổng.*

**20. Tổng số tiền của các hóa đơn theo khách hàng và sắp xếp theo thứ tự giảm dần.**

*Đại số quan hệ không hỗ trợ phép tính tổng.*

**Chú ý**: Một số truy vấn nâng cao, như tính toán trung bình, tổng số, đếm số lượng hoặc các truy vấn phụ thuộc vào thời gian, không thể biểu diễn bằng đại số quan hệ truyền thống. Đại số quan hệ chủ yếu hỗ trợ các phép toán cơ bản như *SELECT* (σ), *PROJECT* (π), *JOIN* (⨝), *DIFFERENCE* (-), và *UNION* (∪), thiếu các phép toán tổng hợp mà SQL cung cấp.

**Bài Cuối Kỳ:** Dưới đây là cách thực hiện truy vấn LINQ to SQL trong C#. Sử dụng lớp DataClasses1DataContext để truy vấn đến các lớp EMPLOYEEs, CUSTOMERs, FEEDBACKs, SUPPLIERs, PRODUCTs, PROMOTIONs, JOB\_TITLEs, CATEGORies, USER\_ACCOUNTs trong cơ sở dữ liệu đã được đổ vào C#. Các lớp này có các thuộc tính đại diện cho các cột trong bảng và các phương thức sử dụng LINQ để thực hiện các câu hỏi như trong SQL.

**1. Tạo lớp và kết nối CSDL với LINQ to SQL**

Đầu tiên, thiết lập LINQ to SQL bằng cách sử dụng DataContext để ánh xạ các bảng từ SQL Server vào các lớp trong C#.

**Thiết lập DataContext trong LINQ to SQL:**

1. **Tạo DataContext:** Trong Visual Studio, tạo một lớp DataClasses1DataContext.
2. **Thêm Bảng vào DataContext:** Thên các bảng Catagory, Customer, Employee, Feedback, Incoice, Job\_Title, Product, Promotion, Supplier, User\_Account. Visual Studio sẽ tự động tạo các lớp tương ứng dựa trên bảng trong SQL Server.

**2. Chương trình chính thực thi các truy vấn**

Sau khi thiết lập DataContext, sẽ tạo các truy vấn tương ứng với các câu truy vấn SQL đã đưa ra.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace StoreCircleK  {  public class ManagementStoreCircleK  {  static void Main(string[] args)  {  DataClasses1DataContext dcdc = new DataClasses1DataContext();  //20 Câu lệnh với LinQ  //1. Lấy ra thông tin nhân viên của Circle K (tên, giới tính, ngày sáng năm sinh, địa chỉ ở)  Console.WriteLine("Cau 1: Thong tin nhan vien");  var thongtin = dcdc.EMPLOYEEs.Select(x => new { x.HoTen, x.GioiTinh, x.NgaySinh, x.DiaChi });  foreach (var t in thongtin)  {  Console.WriteLine($"Ten: {t.HoTen}, Gioi Tinh: : {t.GioiTinh}, Ngay Sinh: {t.NgaySinh}, Đia Chi: {t.DiaChi}");  }  Console.WriteLine("-----------------------------------------------");  //2. Tìm nhân viên bán hàng  Console.WriteLine("Cau 2: Nhan vien ban hang");  var nhanvienbanhang = dcdc.EMPLOYEEs.Where(x => x.MaCV == 1).Select(x => new { x.HoTen });  foreach ( var t in nhanvienbanhang)  {  Console.WriteLine($"{t.HoTen}");  }  Console.WriteLine("-----------------------------------------------");  //3. lấy ra tên và số điện thoại của khách hàng có số lượt mua trên 5  Console.WriteLine("Cau 3: tim khach hang co lan mua tren 5: ");  var khachhang = dcdc.CUSTOMERs.Where(x => x.SoLanMua > 5).Select(x => new { x.HoTen, x.SDT });  foreach ( var t in khachhang)  {  Console.WriteLine($"{t.HoTen} - {t.SDT}");  }  Console.WriteLine("-----------------------------------------------");  //4. lấy ra lời phản hồi của khách hàng có mã khách hàng là 3 và lượt đánh giá từ 4 trở lên  Console.WriteLine("Cau 4: MaKH = 3 va danh gia từ 4 tro len: ");  var makh\_danhgia = dcdc.FEEDBACKs.Where(x => x.MaKH == 3 && x.SoDiemDanhGia >= 4).Select(x => new { x.NgayPhanHoi, x.NoiDung });  foreach(var t in makh\_danhgia)  {  Console.WriteLine($"{t.NgayPhanHoi} - {t.NoiDung}");  }  Console.WriteLine("-----------------------------------------------");  //5. lấy tên của khách hàng đã đánh giá 5 sao  Console.WriteLine("Cau 5: Ten Khach hang danh gia 5 sao: ");  var danhgia = dcdc.FEEDBACKs.Where(x => x.SoDiemDanhGia == 5).Select(x => new { x.MaKH });  foreach ( var t in danhgia)  {  var ten = dcdc.CUSTOMERs.Where(s => s.MaKH == t.MaKH).Select(s => new { s.HoTen, s.SDT });  foreach(var r in ten)  {  Console.WriteLine($"{r.HoTen} - {r.SDT}");  }  }  Console.WriteLine("-----------------------------------------------");  //6. In ra những hóa đơn kèm theo tên của người mua  Console.WriteLine("Cau 6: Ten va so tien: ");  var tien = dcdc.INVOICEs.Select(x => new { x.MaKH, x.TongSoTien });  foreach( var t in tien)  {  var ten = dcdc.CUSTOMERs.Where(s => s.MaKH == t.MaKH).Select(x => new { x.HoTen });  foreach ( var r in ten)  {  Console.WriteLine($"{r.HoTen} - {t.TongSoTien}");  }  }  Console.WriteLine("-----------------------------------------------");  //7.tìm những sản phẩm của Vinamilk có đơn giá trên 25000  Console.WriteLine("Cau 7: San Pham cua Vinamilk co don gia tren 25000");  var nhacungcap = dcdc.SUPPLIERs.Where(x => x.TenNCC.Equals("Vinamilk")).Select(y => new { y.MaNCC});  foreach( var t in nhacungcap)  {  var sanpham = dcdc.PRODUCTs.Where(e => e.MaNCC == t.MaNCC && e.DonGia > 25000).Select(q => new { q.TenSP, q.DonGia });  foreach( var r in sanpham)  {  Console.WriteLine($"{r.TenSP} - {r.DonGia}");  }  }  Console.WriteLine("-----------------------------------------------");  //8. Đếm số lượng nhân viên trong Circle K  Console.WriteLine("Cau 8: So luong nhan vien trong Circle K ");  int soluongnhanvien = dcdc.EMPLOYEEs.Count();  Console.WriteLine(soluongnhanvien);  Console.WriteLine("-----------------------------------------------");  //9. Tính tổng sản phẩm của mỗi nhà cung cấp  Console.WriteLine("Cau 9: Tong san pham cua moi nha cung cap");  var tongsanpham = dcdc.PRODUCTs.GroupBy(x => x.MaNCC).Select(y => new { mancc = y.Key, tong = y.Count() });  foreach( var t in tongsanpham)  {  var tenncc = dcdc.SUPPLIERs.Where(v => v.MaNCC == t.mancc).Select(d => new { d.TenNCC });  foreach( var r in tenncc)  {  Console.WriteLine($"{r.TenNCC} - {t.tong}");  }  }  Console.WriteLine("-----------------------------------------------");  //10. tính tổng số tiền trung bình của các khách hàng  Console.WriteLine("Cau 10: Tinh tong so tien trung binh cua cac khach hang: ");  double tong\_invoice = (double) dcdc.INVOICEs.Sum(s => s.TongSoTien);  int soluong = dcdc.CUSTOMERs.Count();  Console.WriteLine(tong\_invoice/soluong);  Console.WriteLine("-----------------------------------------------");  //11. Lấy ra những phiếu khuyến mãi được áp dụng trong thời gian hiện tại  Console.WriteLine("Cau 11: khuyen mai dang dien ra");  var khuyenmai = dcdc.PROMOTIONs  .Where(x => x.ThoiGianBatDau <= DateTime.Now && x.ThoiGianKetThuc >= DateTime.Now)  .Select(x => new { x.NoiDung, x.MucGiamGia, x.ThoiGianBatDau, x.ThoiGianKetThuc });  foreach (var km in khuyenmai)  {  Console.WriteLine($"Khuyen Mai: {km.NoiDung}, Giam Gia: {km.MucGiamGia}%, Bat Dau: {km.ThoiGianBatDau}, Ket Thuc: {km.ThoiGianKetThuc}");  }  Console.WriteLine("-----------------------------------------------");  //12. tìm sản phẩm hết hạn trong 30 ngày tính từ ngày thời điểm hiện tại  Console.WriteLine("Cau 12: San pham het han trong 30 ngay");  var today = DateTime.Now;  var sapHetHan = dcdc.PRODUCTs  .Where(x => x.HanSuDung <= today.AddDays(30))  .Select(x => new { x.TenSP, x.HanSuDung, x.NoiSanXuat });  foreach (var sp in sapHetHan)  {  Console.WriteLine($"Ten SP: {sp.TenSP}, Han SD: {sp.HanSuDung:d}, Noi SX: {sp.NoiSanXuat}");  }  Console.WriteLine("-----------------------------------------------");  //13. Với mỗi Tên Công việc lấy ra tên của từng nhân viên trong đó  Console.WriteLine("Cau 13: Lay ra nhan vien cua tung cong viec");  var congviec = dcdc.JOB\_TITLEs.Select(x => new { x.TenCV, x.MaCV });  foreach (var job in congviec)  {  Console.WriteLine($"{job.TenCV}");  var nhanvien = dcdc.EMPLOYEEs.Where(s => s.MaCV == job.MaCV).Select(e => new { e.HoTen });  foreach ( var nv in nhanvien)  {  Console.WriteLine($"{nv.HoTen}");  }  }  Console.WriteLine("-----------------------------------------------");  //14. Với mỗi nhà cung cấp tính tổng đơn giá của các sản phẩm của nhà cung cấp  Console.WriteLine("Cau 14: Don gia cua nha cung cap.");  var cungcap = dcdc.PRODUCTs.GroupBy(x => x.MaNCC).Select(y => new { mancc = y.Key, tongdongia = y.Sum(r => r.DonGia)});  var tennccap = dcdc.SUPPLIERs.Select(x => new { x.TenNCC, x.MaNCC });  foreach (var supplier in tennccap)  {  foreach (var cung in cungcap)  {  if(supplier.MaNCC == cung.mancc)  {  Console.WriteLine($"{supplier.TenNCC} - {cung.tongdongia}");  break;  }  }  }  Console.WriteLine("-----------------------------------------------");  //15. lấy ra sản phẩm thuộc nhóm đồ uống và sắp xếp theo đơn giá giảm dần  Console.WriteLine("Cau 15: Sam phan do uong.");  var noibang = dcdc.PRODUCTs.Join(dcdc.CATEGORies, a => a.MaNhom, b => b.MaNhom, (a, b) => new { a.TenSP, b.TenNhom, a.DonGia }).Where(s => s.TenNhom.Equals("Đo uong")).OrderByDescending(y => y.DonGia).Select(d => new { d.TenSP, d.DonGia });  foreach (var supplier in noibang)  {  Console.WriteLine($"{supplier.TenSP} - {supplier.DonGia}");  }  Console.WriteLine("-----------------------------------------------");  //16. lấy ra sản phẩm thuộc nhóm đồ ăn và sắp xếp theo đơn giá tăng dần  Console.WriteLine("Cau 16: Sam phan do an.");  var noibang\_A = dcdc.PRODUCTs.Join(dcdc.CATEGORies, a => a.MaNhom, b => b.MaNhom, (a, b) => new { a.TenSP, b.TenNhom, a.DonGia }).Where(s => s.TenNhom.Equals("Đo an")).OrderBy(y => y.DonGia).Select(d => new { d.TenSP, d.DonGia });  foreach (var supplier in noibang\_A)  {  Console.WriteLine($"{supplier.TenSP} - {supplier.DonGia}");  }  Console.WriteLine("-----------------------------------------------");  //17. lấy ra thông tin khách hàng đã đánh giá và sắp xếp tăng dần theo số lần mua  Console.WriteLine("Cau 17: Thong tin khach hang danh gia va duoc sap xep theo so lan mua");  var tt\_khachhang = dcdc.CUSTOMERs.Join(dcdc.FEEDBACKs, tt => tt.MaKH, b => b.MaKH, (tt, b) => new { tt.HoTen, tt.SoLanMua, b.NoiDung, b.SoDiemDanhGia }).Where(s => s.SoDiemDanhGia != null).OrderBy(y => y.SoLanMua).Select(d => new { d.HoTen, d.SoLanMua, d.NoiDung });  foreach (var tt in tt\_khachhang)  {  Console.WriteLine($"{tt.HoTen} - {tt.NoiDung} - {tt.SoLanMua}");  }  Console.WriteLine("-----------------------------------------------");  //18. lấy ra thông tin khách hàng không đánh giá của hàng Circle K  Console.WriteLine("Cau 18: Thong tin khach hang chua danh gia Circle K");  var chua\_phanhoi = from a in dcdc.CUSTOMERs join b in dcdc.FEEDBACKs on a.MaKH equals b.MaKH into ordergroup from b in ordergroup.DefaultIfEmpty() where (b.SoDiemDanhGia == null) select new { a.HoTen, a.SDT};  foreach (var t in chua\_phanhoi)  {  Console.WriteLine($"{t.HoTen} - {t.SDT}");  }  Console.WriteLine("-----------------------------------------------");  //19. lấy ra thông tin khách hàng mua đơn dưới 100k kèm theo lời phải hồi  Console.WriteLine("Cau 19: Thong tin khach hang mua duoi 100000 va kem theo loi phan hoi");  var donduoi100 = dcdc.INVOICEs.GroupBy(t => t.MaKH).Select(s => new {makh = s.Key, tongtien = s.Sum(r => r.TongSoTien)});  foreach (var s in donduoi100)  {  if(s.tongtien < 100000)  {  var thongtinkhachhang = dcdc.CUSTOMERs.Where(x => x.MaKH == s.makh).Select(y => new { y.HoTen });  foreach (var o in thongtinkhachhang)  {  var phanhoitukhachhang = dcdc.FEEDBACKs.Where(n => n.MaKH == s.makh).Select(a => new { a.NoiDung });  foreach (var q in phanhoitukhachhang)  {  Console.WriteLine($"{o.HoTen} - {q.NoiDung} - {s.tongtien}");  }  }  }  }  Console.WriteLine("-----------------------------------------------");  //20. Tính tổng số tiền của các hóa đơn theo khách hàng và sắp xếp theo thứ tự giảm dần của tổng tiền  Console.WriteLine("Cau 20: Tong tien hoa don theo khach hang duoc sap xep giam dan");  var tongTienTheoKhachHang = dcdc.INVOICEs.GroupBy(hd => hd.MaKH).Select(g => new{MaKH = g.Key, TongTien = g.Sum(hd => hd.TongSoTien) }).OrderByDescending(s => s.TongTien);  foreach (var r in tongTienTheoKhachHang)  {  var khachHang = dcdc.CUSTOMERs.Where(kh => kh.MaKH == r.MaKH).Select(x => new { x.HoTen, x.SDT });  foreach(var o in khachHang)  {  Console.WriteLine($"{o.HoTen} - {o.SDT} - {r.TongTien}");  }  }  Console.WriteLine("-----------------------------------------------");  Console.ReadKey();  }  }  } |

**3. Giải thích chương trình**

1. **Kết nối LINQ to SQL:** Sử dụng DataClasses1DataContext để kết nối với SQL Server và thực hiện các truy vấn.
2. **Lấy ra thông tin nhân viên của Circle K (tên, giới tính, ngày sáng năm sinh, địa chỉ ở)**

|  |
| --- |
| Console.WriteLine("Cau 1: Thong tin nhan vien");  var thongtin = dcdc.EMPLOYEEs.Select(x => new { x.HoTen, x.GioiTinh, x.NgaySinh, x.DiaChi });  foreach (var t in thongtin)  {  Console.WriteLine($"Ten: {t.HoTen}, Gioi Tinh: : {t.GioiTinh}, Ngay Sinh: {t.NgaySinh}, Đia Chi: {t.DiaChi}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc lấy thông tin nhân viên của Circle K. Tạo một biến **thongtin** trong đó lấy ra thông tin của những nhân viên bằng lệnh **Select** từ **EMPLOYEEs** thông qua dcdc. Sau đó khởi tạo vòng **foreach** để in ra thông tin nhân viên gồm có **HoTen, GioiTInh, NgaySinh, DiaChi**.

1. **Tìm nhân viên bán hàng**

|  |
| --- |
| Console.WriteLine("Cau 2: Nhan vien ban hang");  var nhanvienbanhang = dcdc.EMPLOYEEs.Where(x => x.MaCV == 1).Select(x => new { x.HoTen });  foreach ( var t in nhanvienbanhang)  {  Console.WriteLine($"{t.HoTen}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ này thực hiện việc tìm nhân viên bán hàng. Trong này nhân viên bán hàng sẽ có mã **MaCV = 1**, ta sử dụng câu lệnh **Where** để tìm những nhân viên nào có **MaCV = 1** từ bàng **EMPLOYEEs**, nếu tìm thấy ra lấy ra **HoTen**, sau đó sẽ chạy vòng **foreach** để thực hiện việc in ra tên nhân viên bán hàng.

1. **lấy ra tên và số điện thoại của khách hàng có số lượt mua trên 5**

|  |
| --- |
| Console.WriteLine("Cau 3: tim khach hang co lan mua tren 5: ");  var khachhang = dcdc.CUSTOMERs.Where(x => x.SoLanMua > 5).Select(x => new { x.HoTen, x.SDT });  foreach ( var t in khachhang)  {  Console.WriteLine($"{t.HoTen} - {t.SDT}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ này thực hiện việc lấy ra tên và số điện thoại của những khách hàng có số lần mua trên 5. Trong đây ta sử dụng cậu lênh **Where** để dò tìm khách hàng có **SoLanMua** trên 5 từ bảng **CUSTOMERs**, khi tìm được ta sẽ lấy ra thông tin về **HoTen, SDT**, sâu có ta sẽ sử dụng **foreach** để thực hiện việc in thông tin khách hàng.

1. **lấy ra lời phản hồi của khách hàng có mã khách hàng là 3 và lượt đánh giá từ 4 trở lên**

|  |
| --- |
| Console.WriteLine("Cau 4: MaKH = 3 va danh gia từ 4 tro len: ");  var makh\_danhgia = dcdc.FEEDBACKs.Where(x => x.MaKH == 3 && x.SoDiemDanhGia >= 4).Select(x => new { x.NgayPhanHoi, x.NoiDung });  foreach(var t in makh\_danhgia)  {  Console.WriteLine($"{t.NgayPhanHoi} - {t.NoiDung}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ này thực hiện lấy lời phản hồi của khách hàng có **MaKH = 3** và khách hàng đó có đánh giá trên 4 hay không. Để thực hiện được ra sử dụng câu lệnh **Where** để lọc ra những khách hàng có **MaKH = 3** và **SoDiemDanhGia** được lấy từ bảng **FEEDBACKs** của khách hàng đó có từ 4 trở lên không, nếu tìm thấy ta sẽ in ra **NgayPhanhoi** và **NoiDung** của khách hàng đó.

1. **lấy tên của khách hàng đã đánh giá 5 sao**

|  |
| --- |
| var danhgia = dcdc.FEEDBACKs.Where(x => x.SoDiemDanhGia == 5).Select(x => new { x.MaKH });  foreach ( var t in danhgia)  {  var ten = dcdc.CUSTOMERs.Where(s => s.MaKH == t.MaKH).Select(s => new { s.HoTen, s.SDT });  foreach(var r in ten)  {  Console.WriteLine($"{r.HoTen} - {r.SDT}");  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện lấy ra tên của những khách hàng đã đánh giá cho Circle K 5 sao. Trong câu lệnh này sử dụng câu lệnh **where** để tìm ra khách hàng đã đánh giá 5 sao từ bảng **FEEDBACKs** và đồng thời lấy ra **MaKH**. Sau đó ta thực hiện chạy vòng **foreach** và bắt đầu so sánh từ bảng **CUSTOMERs** lấy ra **MaKH** so sánh với **MaKH** từ **FEEDBACKs** nếu bằng nhau thì từ bảng **CUSTOMERs** ta lấy ra **HoTen** và **SDT** của khách hàng tiếp theo ta sẽ chạy **foreach** để thực hiện việc in ra **HoTen** và **SDT** vừa lấy ra.

1. **In ra những hóa đơn kèm theo tên của người mua**

|  |
| --- |
| Console.WriteLine("Cau 6: Ten va so tien: ");  var tien = dcdc.INVOICEs.Select(x => new { x.MaKH, x.TongSoTien });  foreach( var t in tien)  {  var ten = dcdc.CUSTOMERs.Where(s => s.MaKH == t.MaKH).Select(x => new { x.HoTen });  foreach ( var r in ten)  {  Console.WriteLine($"{r.HoTen} - {t.TongSoTien}");  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ này thực hiện việc in ra hóa đơn của khách hàng và kèm theo tên của khách hàng. Trong câu truy vấn trên để in ra được hóa đơn của khách hàng ta phải truy cập đến bảng **INVOICEs** để lấy ra **MaKH** và **TongSoTien**. Sau khi lấy ta chạy vòng **foreach**, ứng với mỗi **MaKH** ta sử dụng câu lệnh **where** để dò tìm **MaKH** từ bảng **CUSTOMERs** nếu có tồn tại ta lấy ra họ tên. Và chạy vòng **foreach** để in ra **HoTen** kèm theo **TongSoTien** của mỗi khách hàng.

1. **tìm những sản phẩm của Vinamilk có đơn giá trên 25000**

|  |
| --- |
| Console.WriteLine("Cau 7: San Pham cua Vinamilk co don gia tren 25000");  var nhacungcap = dcdc.SUPPLIERs.Where(x => x.TenNCC.Equals("Vinamilk")).Select(y => new { y.MaNCC});  foreach( var t in nhacungcap)  {  var sanpham = dcdc.PRODUCTs.Where(e => e.MaNCC == t.MaNCC && e.DonGia > 25000).Select(q => new { q.TenSP, q.DonGia });  foreach( var r in sanpham)  {  Console.WriteLine($"{r.TenSP} - {r.DonGia}");  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc tìm các sản phẩm của nhà cung cấp **Vinalmilk** trong của hàng Circle K và những sản phẩm đó phải có giá trên 25000. Để thực hiện được câu truy vấn trên ta sử dụng câu lệnh **where** và tìm trong **TenNCC** của bảng **SUPPLIERs** tìm “**Vinamilk**” nếu tìm thấy ta sẽ lấy **MaNCC**. Sau đó sẽ chạy vòng **foreach** lấy **MaNCC** vừa tìm được của **Vinamilk** ta sử dụng câu lệnh **where** để tìm trong bảng **PRODUCTs** những sản phẩm có **MaNCC** trùng với **MaNCC** của **Vinamilk**. Nếu tìm thấy sản phẩm như vậy ta sẽ lấy **TenSP** và **DonGia** rồi chạy vòng **foreach** để in ra.

1. **Đếm số lượng nhân viên trong Circle K**

|  |
| --- |
| Console.WriteLine("Cau 8: So luong nhan vien trong Circle K ");  int soluongnhanvien = dcdc.EMPLOYEEs.Count();  Console.WriteLine(soluongnhanvien);  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc đến số lượng nhân viên trong của hàng Circle K. Đế thực hiện được thì ra chỉ cần tạo một biến **soluongnhanvien** thực hiện việc đếm bằng hàm **Count** và ta đến từ bằng **EMPLOUEEs**.

1. **Tính tổng sản phẩm của mỗi nhà cung cấp**

|  |
| --- |
| Console.WriteLine("Cau 9: Tong san pham cua moi nha cung cap");  var tongsanpham = dcdc.PRODUCTs.GroupBy(x => x.MaNCC).Select(y => new { mancc = y.Key, tong = y.Count() });  foreach( var t in tongsanpham)  {  var tenncc = dcdc.SUPPLIERs.Where(v => v.MaNCC == t.mancc).Select(d => new { d.TenNCC });  foreach( var r in tenncc)  {  Console.WriteLine($"{r.TenNCC} - {t.tong}");  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc tính toán tổng số sản phẩm của mỗi nhà cung cấp đã cung cấp cho cửa hàng Circle K. Để thực hiện được ta ban đầu phải sử dụng lệnh **GroupBy** để gom nhóm theo **MaNCC** của bảng **PRODUCTs** và thực hiện việc tính tổng số sản phẩm bằng cách dùng hàm **Count** để đếm số sản phẩm của mỗi nhà cung cấp. Sau đó sẽ chạy vòng **foreach**, tạo một biến **tenncc** có sử dụng lệnh **where** ta thực hiện việc so sánh **mancc** với **MaNCC** từ bảng **SUPPLIERs** để tìm ra **TenNCC**. Ta chạy thêm một vòng **foreach** để in thông tin **TenNCC** kèm theo số lượng sản phẩm đã đếm.

1. **tính tổng số tiền trung bình của các khách hàng**

|  |
| --- |
| Console.WriteLine("Cau 10: Tinh tong so tien trung binh cua cac khach hang: ");  double tong\_invoice = (double) dcdc.INVOICEs.Sum(s => s.TongSoTien);  int soluong = dcdc.CUSTOMERs.Count();  Console.WriteLine(tong\_invoice/soluong);  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc tính toán số tiền trung bình của cửa hàng Circle K dựa trên số lượng khách mua hàng. Để thực hiện được ta ban đầu tạo một biến **tong\_invoice** sau đó gọi hàm Sum từ bảng **INVOICEs** và ta tính tổng tiền từ cột **TongSoTien**, ta phải ép kiểu nó về double vì **tong\_invoice** của ta đang khai áo nó thuộcc double. Tiếp theo ta tạo thêm một biến **soluong** để lấy số lượng kháng hàng. Sau sùng ta in ra số tiền trung bình của cửa hàng.

1. **Lấy ra những phiếu khuyến mãi được áp dụng trong thời gian hiện tại**

|  |
| --- |
| Console.WriteLine("Cau 11: khuyen mai dang dien ra");  var khuyenmai = dcdc.PROMOTIONs  .Where(x => x.ThoiGianBatDau <= DateTime.Now && x.ThoiGianKetThuc >= DateTime.Now)  .Select(x => new { x.NoiDung, x.MucGiamGia, x.ThoiGianBatDau, x.ThoiGianKetThuc });  foreach (var km in khuyenmai)  {  Console.WriteLine($"Khuyen Mai: {km.NoiDung}, Giam Gia: {km.MucGiamGia}%, Bat Dau: {km.ThoiGianBatDau}, Ket Thuc: {km.ThoiGianKetThuc}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc lấy ra những chương trình khuyến mãi được áp dụng trong thời gian hiện tại. Để thực hiện được ta cần thấy thời gian bắt đầu và thời gian kết thúc của chương trình khuyến mãi từ bảng **PROMOTIONs**. Ta sử dụng **DataTime.Now** để xem thời gian hiện tại có nằm trong thời gian sự kiện của chương trình khuyến mãi hay không. Nếu có thì ta sẽ lấy từ bảng **PROMOTIONs** lấy ra **NoiDung, MucGiamGia, ThoiGianBatDau, ThoiGianKetThuc**. Sau khi lấy ta sẽ chạy **foreach** để in thông tin chương trình khuyến mãi còn thời gian hiệu lực tính đếm thời điểm hiện tại.

1. **tìm sản phẩm hết hạn trong 30 ngày tính từ ngày thời điểm hiện tại**

|  |
| --- |
| Console.WriteLine("Cau 12: San pham het han trong 30 ngay");  var today = DateTime.Now;  var sapHetHan = dcdc.PRODUCTs  .Where(x => x.HanSuDung <= today.AddDays(30))  .Select(x => new { x.TenSP, x.HanSuDung, x.NoiSanXuat });  foreach (var sp in sapHetHan)  {  Console.WriteLine($"Ten SP: {sp.TenSP}, Han SD: {sp.HanSuDung:d}, Noi SX: {sp.NoiSanXuat}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh trên thực hiện việc tìm sản phẩn hết hạn trong vòng 30 ngày và tính từ thời điểm hiện tại. Để thực hiện ta sẽ lấy ra ngày hiện tại và gán vào biến **today**. Sau đó ta lấy ra **HanSuDung** từ bảng **PRODUCTs** đi kiếm tra với ngày hiện tại và cộng thêm 30 ngày dùng **AddDay(30)**. Nếu như thõa mãn điều kiện trên thì ta lấy ra **TenSP, HanSuDung, NoiSanXuat** và thực hiện việc in ra bằng **foreach**.

1. **Với mỗi Tên Công việc lấy ra tên của từng nhân viên trong đó**

|  |
| --- |
| Console.WriteLine("Cau 13: Lay ra nhan vien cua tung cong viec");  var congviec = dcdc.JOB\_TITLEs.Select(x => new { x.TenCV, x.MaCV });  foreach (var job in congviec)  {  Console.WriteLine($"{job.TenCV}");  var nhanvien = dcdc.EMPLOYEEs.Where(s => s.MaCV == job.MaCV).Select(e => new { e.HoTen });  foreach ( var nv in nhanvien)  {  Console.WriteLine($"{nv.HoTen}");  }  } |

Câu lệnh LINQ trên thực hiện việc với mỗi công việc lấy ra tên của nhân viên làm công việc đó. Để thực hiện được ta dùng bảng **Job\_Title** lấy ra **TenCV, MaCV**. Sau đó chạy vòng **foreach** lấy **MaCV** từ trên đi so sánh với **MaCV** từ bảng **EMPLOYEEs** nếu như bằng nhau thì ta lấy ra **HoTen** của nhân viên phụ trách. Cuối cùng là chạy **foreach** để in **HoTen** nhân viên tương ứng với từng công việc.

1. **Với mỗi nhà cung cấp tính tổng đơn giá của các sản phẩm của nhà cung cấp**

|  |
| --- |
| Console.WriteLine("Cau 14: Don gia cua nha cung cap.");  var cungcap = dcdc.PRODUCTs.GroupBy(x => x.MaNCC).Select(y => new { mancc = y.Key, tongdongia = y.Sum(r => r.DonGia)});  var tennccap = dcdc.SUPPLIERs.Select(x => new { x.TenNCC, x.MaNCC });  foreach (var supplier in tennccap)  {  foreach (var cung in cungcap)  {  if(supplier.MaNCC == cung.mancc)  {  Console.WriteLine($"{supplier.TenNCC} - {cung.tongdongia}");  break;  }  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc tính tổng giá tiền của các sản phẩm của mỗi nhà cung cấp. Để thực hiện việc tính toán ta cần **GroupBy** theo **MaNCC** từ bảng **PRODUCTs** và gọi hàm **Sum**, hàm **Sum** nhận tính tổng theo **DonGia**. Sau khi tính toán được số tiền ta chạy vòng **foreach** để thực hiện việc in ra thông tin gồm có **TenNCC** được tra từ bảng **SUPPLIERs** và kèm theo **tongdongia** của mỗi nhà cung cấp.

1. **lấy ra sản phẩm thuộc nhóm đồ uống và sắp xếp theo đơn giá giảm dần**

|  |
| --- |
| Console.WriteLine("Cau 15: Sam phan do uong.");  var noibang = dcdc.PRODUCTs.Join(dcdc.CATEGORies, a => a.MaNhom, b => b.MaNhom, (a, b) => new { a.TenSP, b.TenNhom, a.DonGia }).Where(s => s.TenNhom.Equals("Đo uong")).OrderByDescending(y => y.DonGia).Select(d => new { d.TenSP, d.DonGia });  foreach (var supplier in noibang)  {  Console.WriteLine($"{supplier.TenSP} - {supplier.DonGia}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc tìm những sản phẩm nào thuộc nhóm đồ uống và xắp xếp theo đơn giá giảm dần. Để thực hiện ta cần nối bảng giữa bảng **PRODUCTs** và bảng **CATEGORies** theo điều kiện **a.MaNhom = b.MaNhom** sau đó ra chỉ lấy ra **TenSP, TenNhom và DonGia**. Sau khi lấy ra được ta dùng lệnh **where** để lấy tìm những sản phẩm thuộc đồ uống và sau dùng sẽ gọi lệnh **OrderByDescending** để sắp xếp giảm dần theo đơn giá.

1. **lấy ra sản phẩm thuộc nhóm đồ ăn và sắp xếp theo đơn giá tăng dần**

|  |
| --- |
| Console.WriteLine("Cau 16: Sam phan do an.");  var noibang\_A = dcdc.PRODUCTs.Join(dcdc.CATEGORies, a => a.MaNhom, b => b.MaNhom, (a, b) => new { a.TenSP, b.TenNhom, a.DonGia }).Where(s => s.TenNhom.Equals("Đo an")).OrderBy(y => y.DonGia).Select(d => new { d.TenSP, d.DonGia });  foreach (var supplier in noibang\_A)  {  Console.WriteLine($"{supplier.TenSP} - {supplier.DonGia}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên ý tưởng tương tự câu 15 nhưng có một chút đổi nhỏ. Đó là thay vì tìm đồ uống thì ở đây tìm đồ ăn và sắp xếp tăng dần bằng lệnh **OrderBy.**

1. **lấy ra thông tin khách hàng đã đánh giá và sắp xếp tăng dần theo số lần mua**

|  |
| --- |
| Console.WriteLine("Cau 17: Thong tin khach hang danh gia va duoc sap xep theo so lan mua");  var tt\_khachhang = dcdc.CUSTOMERs.Join(dcdc.FEEDBACKs, tt => tt.MaKH, b => b.MaKH, (tt, b) => new { tt.HoTen, tt.SoLanMua, b.NoiDung, b.SoDiemDanhGia }).Where(s => s.SoDiemDanhGia != null).OrderBy(y => y.SoLanMua).Select(d => new { d.HoTen, d.SoLanMua, d.NoiDung });  foreach (var tt in tt\_khachhang)  {  Console.WriteLine($"{tt.HoTen} - {tt.NoiDung} - {tt.SoLanMua}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc lấy những khách nào hàng đã đánh giá cho Circle K và ta sắp xếp tăng tần theo số lần mua của khách hàng. Để thực hiện ta cần **Join** bảng **CUSTOMERs** và bảng **FEEDBACKs** thông qua **MaKH**. Sau đó ta giữ lại các cột **HoTen, SoLanMua, NoiDung, SoDiemDanhGia**. Sau khi lọc ta dùng lệnh where để giữ lại những khách hàng có **SoDiemDanhGia** không được rỗng thì ra sẽ dùng **OrderBy** để sắp xếp việc in khách hàng theo **SoLanMua** tăng dần.

1. **lấy ra thông tin khách hàng không đánh giá của hàng Circle K**

|  |
| --- |
| Console.WriteLine("Cau 18: Thong tin khach hang chua danh gia Circle K");  var chua\_phanhoi = from a in dcdc.CUSTOMERs join b in dcdc.FEEDBACKs on a.MaKH equals b.MaKH into ordergroup from b in ordergroup.DefaultIfEmpty() where (b.SoDiemDanhGia == null) select new { a.HoTen, a.SDT};  foreach (var t in chua\_phanhoi)  {  Console.WriteLine($"{t.HoTen} - {t.SDT}");  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc lấy ra những khách hàng chưa đánh giá Circle K. Để thực hiện ta cũng **Join** từ bảng **CUSTOMERs** và bảng **FEEDBACKs** thông qua **MaKH** sau đó dùng lệnh **where** để lọc ra cột **SoDiemDanhGia** có giá trị là null thì ta **Select HoTen, SDT**. Sau cùng là chạy vòng **foreach** để in thông tin của khách hàng ra.

1. **lấy ra thông tin khách hàng mua đơn dưới 100k kèm theo lời phải hồi**

|  |
| --- |
| Console.WriteLine("Cau 19: Thong tin khach hang mua duoi 100000 va kem theo loi phan hoi");  var donduoi100 = dcdc.INVOICEs.GroupBy(t => t.MaKH).Select(s => new {makh = s.Key, tongtien = s.Sum(r => r.TongSoTien)});  foreach (var s in donduoi100)  {  if(s.tongtien < 100000)  {  var thongtinkhachhang = dcdc.CUSTOMERs.Where(x => x.MaKH == s.makh).Select(y => new { y.HoTen });  foreach (var o in thongtinkhachhang)  {  var phanhoitukhachhang = dcdc.FEEDBACKs.Where(n => n.MaKH == s.makh).Select(a => new { a.NoiDung });  foreach (var q in phanhoitukhachhang)  {  Console.WriteLine($"{o.HoTen} - {q.NoiDung} - {s.tongtien}");  }  }  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ thực hiện việc lấy ra thông tin những khách hàng mua hàng ở Circle K dưới 100000 VNĐ. Để thực hiện ta dùng **GroupBy** gom nhóm theo **MaKH** của bảng **INVOICEs** và trong đó dùng hàm **Sum** để tính tổng theo **TongSoTien**. Sau khi tính ta chạy vòng **foreach** nếu như số tiền của ta dưới 100000 thì ta truy cập vào bảng **CUSTOMERs** và **FEEDBACKs** để lấy ra **HoTen, NoiDung** của khách hàng. Sau cùng ta sẽ in ra gồm có **HoTen, NoiDung, tongtien**.

1. **Tính tổng số tiền của các hóa đơn theo khách hàng và sắp xếp theo thứ tự giảm dần của tổng tiền**

|  |
| --- |
| Console.WriteLine("Cau 20: Tong tien hoa don theo khach hang duoc sap xep giam dan");  var tongTienTheoKhachHang = dcdc.INVOICEs.GroupBy(hd => hd.MaKH).Select(g => new{MaKH = g.Key, TongTien = g.Sum(hd => hd.TongSoTien) }).OrderByDescending(s => s.TongTien);  foreach (var r in tongTienTheoKhachHang)  {  var khachHang = dcdc.CUSTOMERs.Where(kh => kh.MaKH == r.MaKH).Select(x => new { x.HoTen, x.SDT });  foreach(var o in khachHang)  {  Console.WriteLine($"{o.HoTen} - {o.SDT} - {r.TongTien}");  }  }  Console.WriteLine("-----------------------------------------------"); |

Câu lệnh LINQ trên thực hiện việc với mỗi khách hàng ra tính tổng số tiền trong hóa đơn mà khách hàng đã mua tại Circle K và sắp xếp giảm dần theo tổng tiền. Để thực hiện ta cần **GroupBy** theo **MaKH** của bảng **INVOICEs** và tính tổng tiền bằng hàm **Sum** theo cột **TongSoTien**. Sau đó sẽ sắp xếp giảm dần bằng hàm **OrderByDescending** theo cột **TongTien**. Cuối cùng với mỗi số tiền được tính theo **MaKH** ta in thêm thông tin về **HoTen** và **SDT** của khách hàng từ bảng **CUSTOMERs**.