**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ CƠ SỞ DỮ LIỆU HỆ THỐNG**

**CỬA HÀNG XE ĐẠP**

Giảng viên hướng dẫn: ThS. PHẠM THỊ MIÊN

Sinh viên thực hiện: NGUYỄN NGUYÊN HUY

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :K63

Tp. Hồ Chí Minh, năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**HỌC PHẦN: CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ CƠ SỞ DỮ LIỆU HỆ THỐNG**

**CỬA HÀNG XE ĐẠP**

Giảng viên hướng dẫn: ThS. PHẠM THỊ MIÊN

Sinh viên thực hiện: NGUYỄN NGUYÊN HUY

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :K63

Tp. Hồ Chí Minh, năm 2021

**LỜI CẢM ƠN**

Qua thời gian học tập và rèn luyện tại trường Trường Đại học Giao thông Vận tải phân hiệu tại TP HCM, đến nay em đã được trang bị những kĩ năng, kiến thức cơ bản để có thể hoàn thành được bài tập lớn do giảng viên giao.

Cảm ơn tập thể các thầy cô giáo Bộ môn Công Nghệ Thông Tin và các thầy cô thỉnh giảng đã giảng dạy, luôn quan tâm và không ngần ngại dành thời gian để chỉ bày và giải đáp những thắc mắc của chúng em trong những tiết học và cả những lúc ngoài giờ.

Và cảm ơn thạc sĩ Phạm Thị Miên đã luôn quan tâm nhiệt tình hướng dẫn, giúp đỡ chúng em trong quá trình triển khai và thực hiện bài tập lớn. Cô cũng luôn nhắc nhở, giúp đỡ mỗi khi chúng em gặp khó khăn, nhờ vậy mà em đã hoàn thành bài tập lớn của nhóm mình đúng thời hạn được giao. Nếu không có sự hướng dẫn nhiệt tình của cô thì có lẽ chúng em đã khó có thể thực hiện được bài tập đúng theo mong muốn của mình.

Nhóm chúng em đã bỏ ra nhiều thời gian để tìm hiểu và trang bị thêm kiến thức nhằm phục vụ cho việc thực hiện ý tưởng, nhưng chắc chắn rằng nhóm chúng em sẽ không thể tránh khỏi những sai sót không đáng có vì kiến thức còn hạn chế. Chúng em hi vọng rằng sẽ nhận được những lời góp ý quý báu của cô để chúng em có thể hoàn thiện ý tưởng của nhóm một cách tốt nhất có thể.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***  **Giảng viên hướng dẫn**  **Phạm Thị Miên** |

**BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ**

*[Hình](#_Toc5887)* [1](#_Toc5887) *[. Một số Database Management System](#_Toc5887)* [9](#_Toc5887)

*[Hình](#_Toc18394)* [2](#_Toc18394) *[. Mô hình Database System](#_Toc18394)* [10](#_Toc18394)

*[Hình](#_Toc7797)* [3](#_Toc7797) *[. Lược đồ quan hệ](#_Toc7797)* [14](#_Toc7797)

*[Hình](#_Toc19633)* [4](#_Toc19633) *[. Kiểu liên kết bậc 1](#_Toc19633)* [15](#_Toc19633)

*[Hình](#_Toc5739)* [5](#_Toc5739) *[. Kiểu liên kết bậc 2](#_Toc5739)* [16](#_Toc5739)

*[Hình](#_Toc6394)* [6](#_Toc6394) *[. Kiểu liên kết bậc 3](#_Toc6394)* [16](#_Toc6394)

*[Hình](#_Toc2973)* [7](#_Toc2973) *[. Truy vấn lồng nhau](#_Toc2973)* [20](#_Toc2973)

*[Hình](#_Toc10341)* [8](#_Toc10341) *[. Mô hình ER](#_Toc10341)* [23](#_Toc10341)

*[Hình](#_Toc10589)* [9](#_Toc10589) *[. Sơ lược bài toán](#_Toc10589)* [25](#_Toc10589)

[Hình 10 . Bảng production.brands 32](#_Toc25597)

[Hình 11 . Bảng productions.categories 32](#_Toc7467)

*[Hình](#_Toc16987)* [12](#_Toc16987) *[. Bảng productions.products](#_Toc16987)* [33](#_Toc16987)

*[Hình](#_Toc22663)* [13](#_Toc22663) *[. Bảng productions.stocks](#_Toc22663)* [34](#_Toc22663)

*[Hình](#_Toc14361)* [14](#_Toc14361) *[. Bảng sales.orders](#_Toc14361)* [35](#_Toc14361)

*[Hình](#_Toc9475)* [15](#_Toc9475) *[. Bảng sales.order\_items](#_Toc9475)* [35](#_Toc9475)

*[Hình](#_Toc3210)* [16](#_Toc3210) *[. Bảng sales.staff](#_Toc3210)* [36](#_Toc3210)

*[Hình](#_Toc16028)* [17](#_Toc16028) *[. Bảng sales.stores](#_Toc16028)* [36](#_Toc16028)

*[Hình](#_Toc26280)* [18](#_Toc26280) *[. Bảng sales.customers](#_Toc26280)* [37](#_Toc26280)

**Mục lục**

Lời cảm ơn………………………………………………………………………….……i

Nhận xét của GVHD……………………………………………………………………………..…….ii

Mục lục………………………………………………………………………………….iii

Hình vẽ………………………………………………………………………………….iv

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 8](#_Toc10901)

[1.1 Tổng quan về cơ sở dữ liệu 8](#_Toc18333)

[1.1.1 Cơ sở dữ liệu (Database) 8](#_Toc27035)

[1.1.2 Hệ quản trị CSDL (Database Management System) 9](#_Toc9361)

[1.2 Mô hình thực thể - liên kết ER (Entity - Relationship Model) 11](#_Toc11342)

[1.2.1 Thực thể 11](#_Toc22761)

[1.2.2 Thuộc tính 11](#_Toc32403)

[1.2.3 Mối liên kết 12](#_Toc26191)

[1.2.4 Lược đồ thực thể - liên kết 14](#_Toc21345)

[1.3 Mối liên kết (Entity Relationship) 15](#_Toc21513)

[1.4 Ngôn ngữ truy vấn dữ liệu (Structured Query Language) 18](#_Toc7266)

[1.4.1 Câu lệnh SELECT tổng quát 18](#_Toc16021)

[1.4.2 Mệnh đề SELECT, FROM 18](#_Toc6069)

[1.4.3 Mệnh đề WHERE 19](#_Toc30943)

[1.4.4 Mệnh đề ORDER BY 19](#_Toc19634)

[1.4.5 Mệnh đề GROUP BY 19](#_Toc29814)

[1.4.6 Truy vấn từ nhiều bảng 19](#_Toc14304)

[1.4.7 Truy vấn lồng nhau 20](#_Toc4146)

[1.5 Ràng buộc toàn vẹn 21](#_Toc26490)

[1.5.1 Điều kiện RBTV 21](#_Toc10525)

[1.5.2 Bối cảnh 21](#_Toc5427)

[1.5.3 Bảng tầm ảnh hưởng 21](#_Toc26533)

[CHƯƠNG 2: XÂY DỰNG CƠ SỞ DỮ LIỆU CHO HỆ THỐNG CỦA HÀNG XE ĐẠP 22](#_Toc28449)

[2.1 Mô Tả Bài Toán 22](#_Toc4342)

[2.2 Mô hình ER 23](#_Toc18922)

[2.3 Mô hình quan hệ 24](#_Toc30694)

[2.4 Các ràng buộc 26](#_Toc6554)

[2.5 Bảng tầm ảnh hương của RBTV 27](#_Toc20212)

[2.6 Tạo bảng và chèn dữ liệu 27](#_Toc18227)

[2.6.1 Tạo bảng: 27](#_Toc7807)

[2.6.2 Data 32](#_Toc32179)

[2.7 Truy vấn dữ liệu 37](#_Toc315)

[KẾT LUẬN 45](#_Toc25065)

[TÀI LIỆU THAM KHẢO 46](#_Toc20992)

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

**1.1 Tổng quan về cơ sở dữ liệu**

1.1.1 Cơ sở dữ liệu (Database)

- Cơ sở dữ liệu là một tập hợp có cấu trúc của những dữ liệu có liên quan với nhau

được lưu trữ trong máy tính. Một CSDL biểu diễn một phần của thế giới thực.

CSDL được thiết kế, xây dựng, và lưu trữ với một mục đích xác định, phục vụ cho

một số ứng dụng và người dùng. Tập ngẫu nhiên của các dữ liệu không thể xem là

một CSDL

- Ưu điểm nổi bật của CSDL:

• Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó bảo đảm được tính nhất quán và toàn vẹn dữ liệu.

• Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.

• Khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác nhau.…

- CSDL đặt ra những vấn đề cần giải quyết:

• Tính chủ quyền của dữ liệu: Do tính chia sẻ của CSDL nên chủ quyền của CSDL dễ bị xâm phạm.

• Tính bảo mật và quyền khai thác thông tin: Cần phải có một cơ chế bảo mật và phân quyền hạn khai thác CSDL.

• Tranh chấp dữ liệu: Cần phải có một cơ chế ưu tiên truy nhập dl, cơ chế giải quyết tình trạng khóa chết (DeadLock) trong quá trình khai thác cạnh tranh.

• Đảm bảo dữ liệu khi có sự cố: Khi có sự cố như mất điện đột xuất, đĩa hỏng…

- Các đối tượng sử dụng CSDL:

• Những người sử dụng CSDL không chuyên về lĩnh vực tin học và CSDL.

• Các chuyên viên tin học biết khai thác CSDL.

• Những người quản trị CSDL.

1.1.2 Hệ quản trị CSDL (Database Management System)

- Hệ quản trị CSDL là tập hợp các chương trình cho phép người sử dụng tạo ra và

duy trì CSDL. Một phần mềm hệ thống cho phép định nghĩa, xây dựng và xử lý dữ

liệu

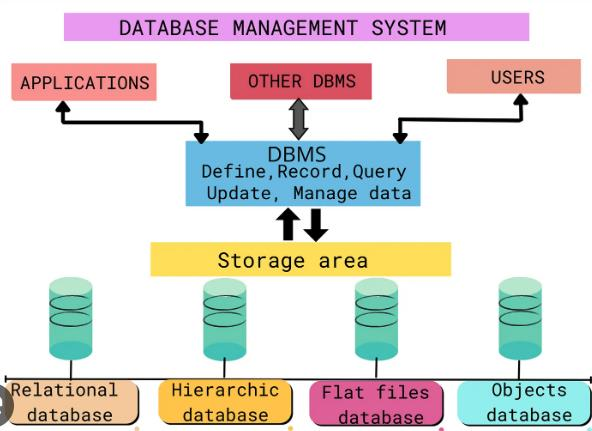
- Một số DBMS: Visual FoxPro, MicroSoft Access, SQL-Server, MySQL, DB2,

Sybase, Paradox, Informix, Oracle…



*Hình 1**. Một số Database Management System*

- Hệ CSDL (Database System)



*Hình 2**. Mô hình Database System*

- Các tính năng của HQT CSDL:

• Kiểm soát được tính dư thừa của dữ liệu

• Chia sẻ dữ liệu

• Hạn chế những truy cập không cho phép

• Cung cấp nhiều giao diện

• Đảm bảo các ràng buộc toàn vẹn

• Khả năng sao lưu dự phòng khi gặp sự cố

• Các tính năng khác

o Chuẩn hóa

o Uyển chuyển

o Giảm thời gian phát triển ứng dụng

o Tính khả dụng

**1.2 Mô hình thực thể - liên kết ER (Entity - Relationship Model)**

Mô hình ER được dùng để thiết kế CSDL ở mức quan niệm. Biểu diễn trừu tượng

cấu trúc của CSDL. Lược đồ thực thể - liên kết (Entity-Relationship Diagram) bao

gồm: tập thực thể (Entity Sets), thuộc tính (Attributes), mối quan hệ (Relationship)

1.2.1 Thực thể

- Một thực thể là một đối tượng của thế giới thực

- Tập hợp các thực thể giống nhau tạo thành 1 tập thực thể

* Ký hiệu: 

- Thực thể yếu: là thực thể phụ thuộc vào sự tồn tại của thực thể khác.

* Ký hiệu: 

1.2.2 Thuộc tính

- Là những đặc tính riêng biệt của tập thực thể

- Ví dụ tập thực thể SACH có các thuộc tính: mã sách, tên sách, giá bán,….

- Là những giá trị nguyên tố: Kiểu chuỗi, kiểu số nguyên, kiểu số thực,..

* Ký hiệu: 

- Các loại thuộc tính:

• Thuộc tính đơn (simple)/đơn trị (single-value)/nguyên tử (Atomic): chỉ có một giá trị trong một thuộc tính của một thực thể. Ví dụ: Thuộc tính “Tên sách” của thực thể SACH

• Thuộc tính tổ hợp (Composite): được kết hợp của một số thành phần. Ví dụ:

Diachi (sonha, duongpho, quan, TP)

• Thuộc tính đa trị (multi-valued): có thể có nhiều giá trị tại một thời điểm. Ví dụ: thuộc tính Trinhdongoaingu của HOCVIEN {Trinhdongoaingu (Anh, Nhat, Trung)}. Ký hiệu: 

• Thuộc tính suy diễn (derived attribute) là giá trị của nó có thể suy ra từ giá trịcủa những thuộc tính khác. Ví dụ: Tuổi (derived attribute) được suy diễn từ Ngaysinh (stored attribute). Ký hiệu: 

• Giá trị rỗng của thuộc tính (Null Values): là thuộc tính có thể có giá trị hoặc không.

• Khoá (key) hay Thuộc tính định danh: là một hoặc một tập các thuộc tính xác định duy nhất một bản ghi trong một thực thể. Thuộc tính khoá hay còn gọi làthuộc tính định danh luôn được gạch chân để phân biệt. Ví dụ: Thuộc tính MaNV xác định tính duy nhất của 1 Nhân viên ( MaNV ).

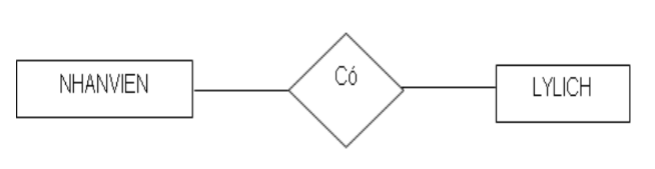
1.2.3 Mối liên kết

- Là sự liên kết giữa 2 hay nhiều tập thực thể

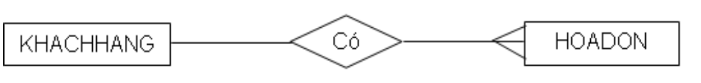
* Ký hiệu: 

- Ràng buộc liên kết gồm:

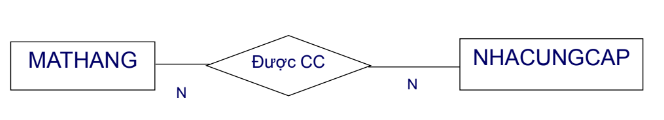
• Liên kết/quan hệ 1-1: là mối quan hệ mà mỗi bản ghi trong thực thể E1 chỉ có nhiều nhất một bản ghi được liên kết trong thực thể E2.



• Liên kết/Quan hệ 1-N: là mối quan hệ mà một bản ghi trong thực thể E1 có thể không liên kết hoặc liên kết với một hay nhiều bản ghi trong thực thể E2, nhưng một bản ghi trong thực thể E2 chỉ liên kết với duy nhất một bản ghi trong thực thể E1.



• Liên kết/Quan hệ M-N: là mối quan hệ mà một bản ghi trong thực thể có thể liên kết với nhiều bản thể trong thực thể khác và ngược lại.



• Lực lượng của mối liên kết: Là số bản ghi lớn nhất và nhỏ nhất của thực thể tham gia vào liên kết đó.



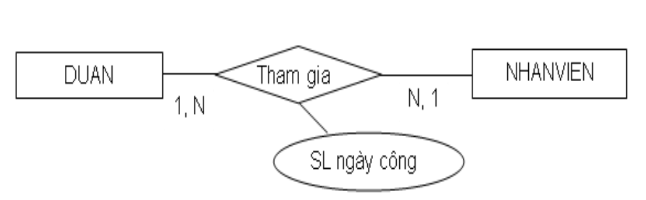
o (0,1) – không hoặc 1

o (1,1) – duy nhất 1

o (0,n) – không hoặc nhiều

o (1,n) – một hoặc nhiều

• Thuộc tính của mối liên kết (Relationship Attribute): Mỗi mối liên kêt cũng có thể có những thuộc tính riêng của chúng.

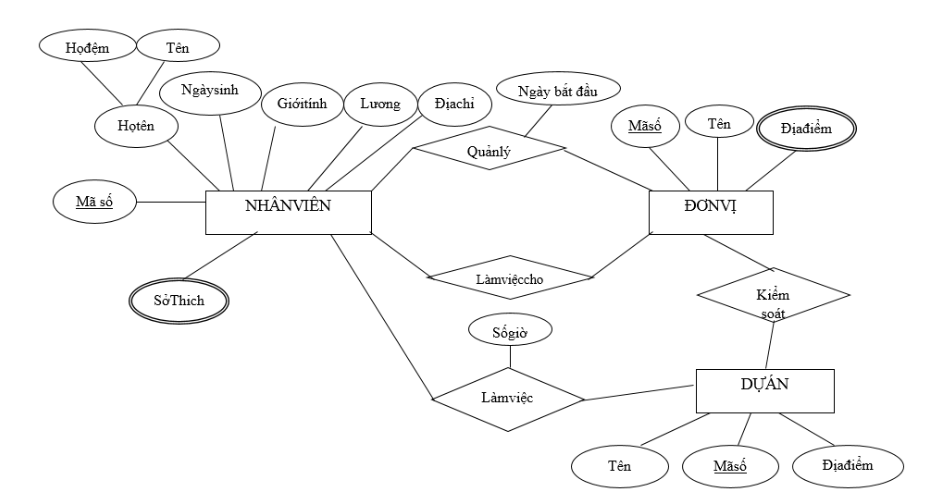


1.2.4 Lược đồ thực thể - liên kết

- Là đồ thị biểu diễn các tập thực thể, thuộc tính và mối quan hệ. Cạnh là đường nối

giữa: tập thực thể và thuộc tính, mối quan hệ và tập thực thể

- Ví dụ:



*Hình 3**. Lược đồ quan hệ*

**1.3 Mối liên kết (Entity Relationship)**

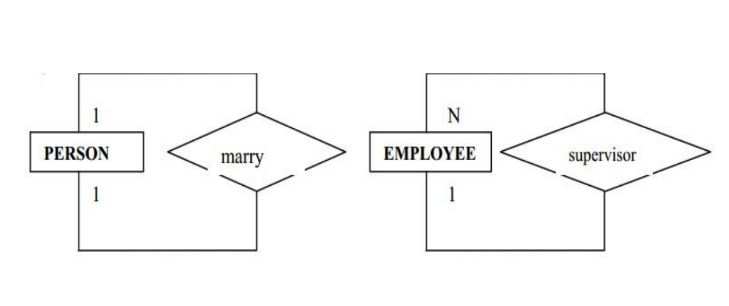
-Là sự liên kết giữa 2 hay nhiều tập thực thể

-Ví dụ giữa tập thực thể NHANVIEN và

* PHONGBAN có các liên kết
* Một nhân viên thuộc một phòng ban nào đó
* Một phòng ban có một nhân viên làm trưởng phòng
* Một phòng ban có nhiều nhân viên làm việc

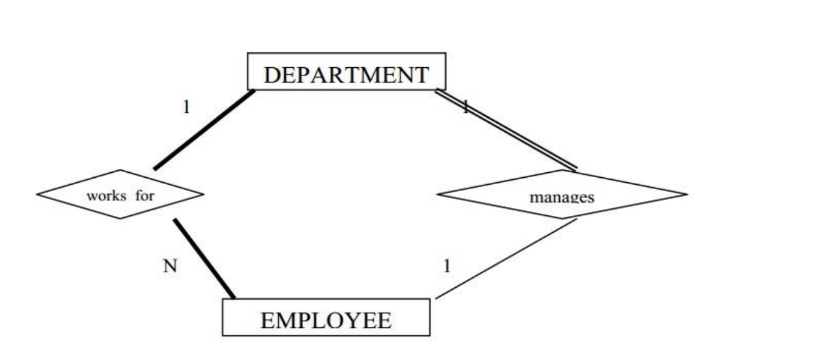
-Bậc của kiểu liên kết: Là số lượng các thực thể tham gia vào liên kết.

* Kiểu liên kết bậc 1 (đệ quy) là mối quan hệ giữa cùng 1 kiểu thực thể.



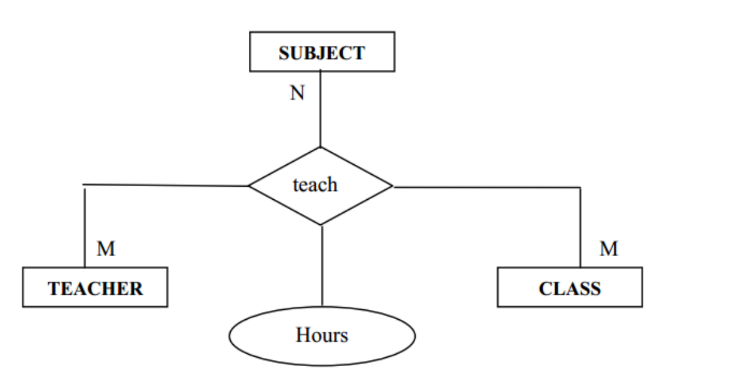
*Hình 4**. Kiểu liên kết bậc 1*

* Kiểu liên kết bậc 2 là mối liên kết giữa hai kiểu thực thể



*Hình 5**. Kiểu liên kết bậc 2*

* Kiểu liên kết bậc 3 là mối liên kết giữa 3 kiểu thực thể

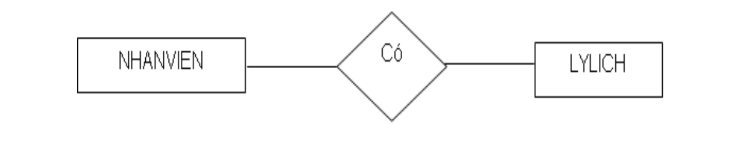


*Hình 6**. Kiểu liên kết bậc 3*

- Ràng buộc liên kết gồm:

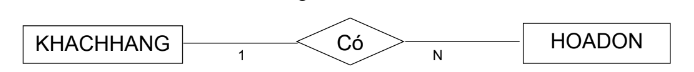
● Liên kết/quan hệ 1-1: là mối quan hệ mà mỗi bản ghi trong thực thể E1 chỉ có nhiều nhất một bản ghi được liên kết trong thực thể E2.

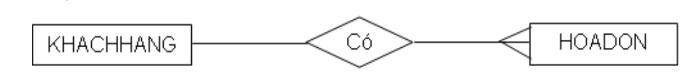
Ví dụ: Mối quan hệ giữa thực thể NHANVIEN và thực thể LYLICH. Mỗi nhân viên chỉ có một lý lịch duy nhất.



● Liên kết/Quan hệ 1-N: là mối quan hệ mà một bản ghi trong thực thể E1 có thể không liên kết hoặc liên kết với một hay nhiều bản ghi trong thực thể E2, nhưng một bản ghi trong thực thể E2 chỉ liên kết với duy nhất một bản ghi trong thực thể E1.

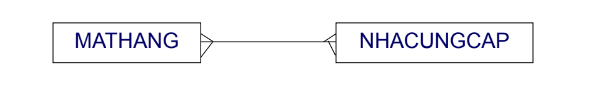
Ví dụ: Mối quan hệ giữa thực thể KHACHHANG và thực thể HOADON. Một khách hàng có thể có một hoặc nhiều hóa đơn, mỗi hóa đơn chỉ thuộc một khách hàng nhất định.

 hoặc:

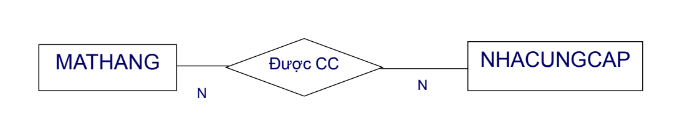


● Liên kết/Quan hệ M-N: là mối quan hệ mà một bản ghi trong thực thể có thể liên kết với nhiều bản thể trong thực thể khác và ngược lại.

Ví dụ: Mối quan hệ giữa thực thể MATHANG và NHACUNCAP. Một mặt hàng có thể do nhiều nhà cung cấp cung cấp và một nhà cung cấp sẽ cung cấp nhiều mặt hàng.



Hoặc:



**1.4 Ngôn ngữ truy vấn dữ liệu (Structured Query Language)**

1.4.1 Câu lệnh SELECT tổng quát

SELECT distinct /\*/danh sách thuộc tính/

<biểu thức>,...

FROM danh\_sách\_các\_quan hệ

[WHERE điều\_kiện]

[GROUP BY danh\_sách\_cột\_cần\_nhóm ]

[HAVING BY điều\_kiện\_nhóm ]

[ORDER BY danh\_sách\_cột\_cần\_sắp\_xếp ]

1.4.2 Mệnh đề SELECT, FROM

- Phát biểu Select với AS: Đặt lại tên Field khi hiển thị kết quả

Cú pháp: Tên cột của bảng AS tên mới

(Nếu tên có chứa các ký tự đặc biệt và/hoặc khoảng trắng thì viết tên đó trong cặp dấu ngoặc vuông)

- Phát biểu Select với TOP N: Cho phép lấy ra chỉ một số mẫu tin nào đó theo 1

tiêu chuẩn nhất định.

- Có thể dùng các phép toán số học +,-,\*,/, ^,%, các hàm tính toán đối với các cột

kiểu số trong mệnh đề SELECT

1.4.3 Mệnh đề WHERE

- WHERE <biểu thức điều kiện>:

• <biểu thức điều kiện>: Biểu thức boolean xác định dòng nàosẽ được rút trích

• Nối các biểu thức: AND, OR, và NOT

• Phép toán: <, >, , ≥, ≠, =, LIKE, BETWEEN,..

1.4.4 Mệnh đề ORDER BY

- Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

ORDER BY <danh sách các cột> [ASC/DESC]

1.4.5 Mệnh đề GROUP BY

- Dùng để nhóm các bản ghi có giá trị giống nhau trên các cột được liệt kê sau từ

khóa GROUP BY.

SELECT <danh sách các cột>

FROM <danh sách các bảng>

[WHERE <điều kiện>]

GROUP BY <tên cột 1>,<tên cột 2>,...

[HAVING <điều kiện>]

1.4.6 Truy vấn từ nhiều bảng

- Khi thông tin cần lấy ra có từ nhiều bảng khác nhau, cần thực hiện truy vấn từ

nhiều bảng

SELECT <Danh sách các cột>

FROM <Danh sách các Bảng>

WHERE <Các điều kiện>

- Các bảng được liên kết với nhau qua phép kết nối của từ khóa FROM hoặc thông

qua điều kiện của từ khóa WHERE của câu lệnh SELECT

• Kết nối bảng qua điều kiện của WHERE:

Tenquanhe1.TTKN=Tenquanhe2.TTKN

• Liên kết bảng với nhau qua phép kết nối của từ khóa FROM → sử dụng phép kết nối nội Inner join:

Quanhe1 Inner join Quanhe2 On Quanhe1.TT1 = Quanhe2.TT2

1.4.7 Truy vấn lồng nhau

- Là những câu lệnh mà trong mệnh đề WHERE hoặc HAVING (thường là trong

mệnh đề WHERE) có chứa thêm một câu lệnh Select khác nữa. Câu lệnh này

thường gặp khi cần sử dụng kết quả của 1 câu truy vấn để làm điều kiện cho 1 câu

truy vấn khác. Khi thực hiện, truy vấn con sẽ được thực hiện trước, rồi lấy kết quả

để thực hiện truy vấn cha



*Hình 7**. Truy vấn lồng nhau*

**1.5 Ràng buộc toàn vẹn**

1.5.1 Điều kiện RBTV

- Điều kiện của ràng buộc toàn vẹn là sự mô tả, và biểu diễn hình thức nội dung của

nó. Có thể được biểu diễn bằng:

• Ngôn ngữ tự nhiên

• Thuật giải (bằng mã giả - Pseudo Code, ngôn ngữ tựa Pascal)

• Ngôn ngữ đại số tập hợp, đại số quan hệ ...

• Các phụ thuộc hàm

- Ví dụ: R1 : “Mỗi hóa đơn có một Số hóa đơn riêng biệt, không trùng với hóa đơn

khác”:

hđ1, hđ2  HOADON, hđ1 ≠ hđ2 → hđ1.Sohoadon ≠ hđ2. Sohoadon.

1.5.2 Bối cảnh

- Là những quan hệ có khả năng bị vi phạm RBTV khi thực hiện các phép cập nhật

- Ví dụ: Trong ví dụ trên, bối cảnh của các ràng buộc toàn vẹn như sau: R1: chỉ là

một quan hệ HÓAĐƠN.

1.5.3 Bảng tầm ảnh hưởng

- Xác định thao tác cập nhật nào cần phải kiểm tra RBTV khi được thực hiện trên

quan hệ bối cảnh

• (+): Vi phạm RBTV

• (-): Không vi phạm RBTV

• +(A): có thể vi phạm RBTV khi sửa trên thuộc tính A

• ( -(\*): Không bị vi phạm vì thao tác không thực hiện được

**CHƯƠNG 2: XÂY DỰNG CƠ SỞ DỮ LIỆU CHO HỆ THỐNG CỦA HÀNG XE ĐẠP**

## 2.1 Mô Tả Bài Toán

Hệ thống quản lí cơ sở dữ liệu của bài toán này được mô tả như sau:

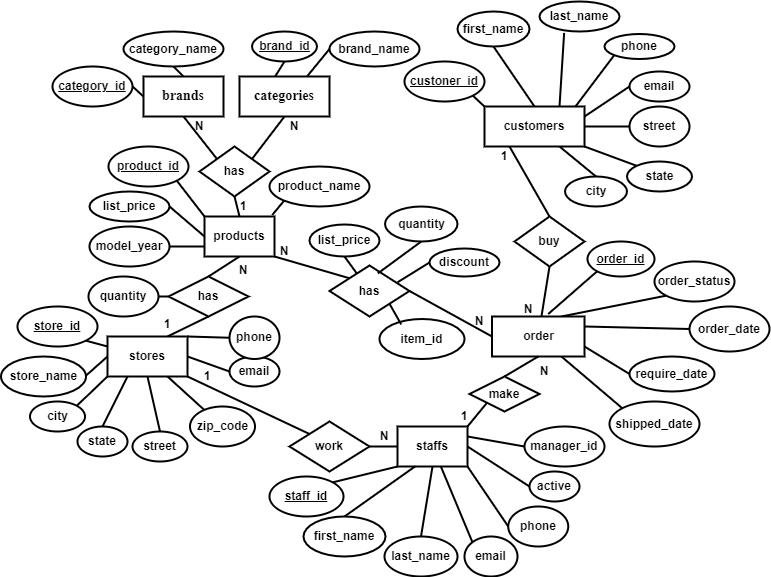
+ Từng loại sản phẩm xe đạp có các thông tin như: mã sản phẩm, tên sản phẩm, năm sản xuất và giá thành. Các thương hiệu (brands) có thể sản xuất nhiều sản phẩm thuộc các danh mục khác nhau. Danh mục sản phẩm có tên danh mục và mã danh mục. Thương hiệu sản phẩm gồm có mã thương hiệu và tên thương hiệu.

+ Khách hàng (customers) có họ, tên, số điện thoại, email, địa chỉ (gồm street, state, city) và mã khách hàng.

+ Thông tin về các cửa hàng (stores) bán xe đạp có các thông tin như tên cửa hàng, số điện thoại, email, địa chỉ và mã cửa hàng. Mỗi cửa hàng có nhiều sản phẩm tồn kho (stocks) và số lượng ứng với từng sản phẩm. Mỗi nhân viên (staffs) gồm cả quản lí làm việc tại duy nhất một cửa hàng có các thông tin như họ, tên, số điện thoại, email. Mỗi cửa hàng có nhiều nhân viên.

+ Đơn hàng được đặt (orders) bởi khách hàng và ứng với một người nhân viên tạo ra đơn hàng đó, thông tin mỗi đơn hàng bao gồm trạng thái, ngày đặt hàng, ngày yêu cầu giao hàng, ngày giao hàng. Các đơn hàng có thông tin chi tiết về các sản phẩm (order\_items) như mã mục đơn hàng cho biết thứ thự sản phẩm trong đơn hàng (Item\_id), số lượng, giá bán và chiết khấu của từng sản phẩm trong đơn hàng đó. Một đơn hàng có nhiều sản phẩm và mỗi loại sản phẩm có thể có trong nhiều đơn hàng khác nhau.

## 2.2 Mô hình ER



*Hình 8**. Mô hình ER*

## 2.3 Mô hình quan hệ

production.categories ( category\_id, category\_name)

production.brands ( brand\_id, brand\_name)

production.products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

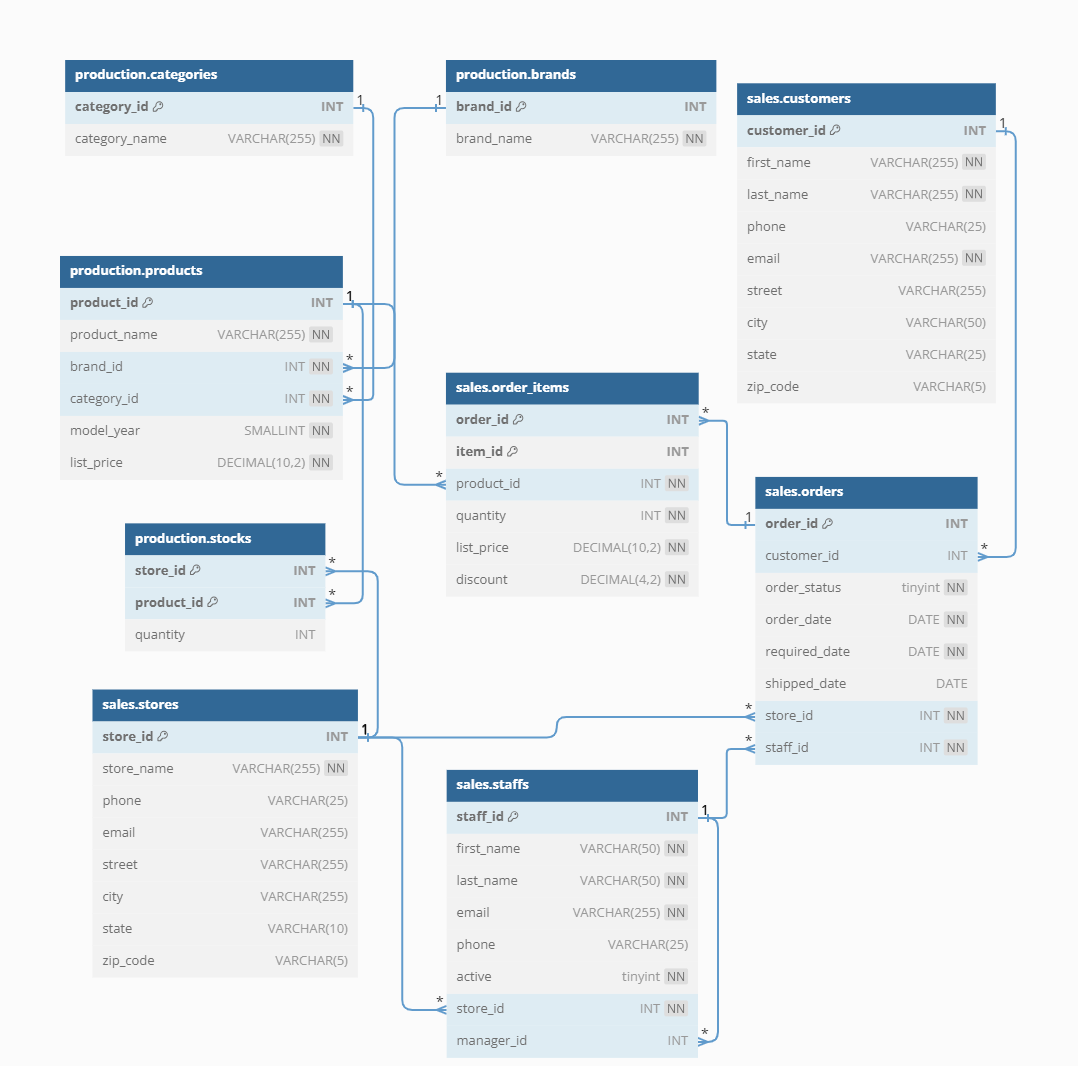
production.stocks ( store\_id, product\_id, quantity)

sales.customers ( customer\_id, first\_name, last\_name, phone, email, street, city, state, zip\_code)

sales.stores ( store\_id, store\_name, phone, email, street, city, state, zip\_code) sales.staffs ( staff\_id, first\_name, last\_name, email, phone, active, store\_id, manager\_id )

sales.orders ( order\_id, customer\_id, order\_status, order\_date, required\_date, shipped\_date, store\_id, staff\_id)

sales.order\_items ( order\_id, item\_id, product\_id, quantity, list\_price, discount)



*Hình 9**. Sơ lược bài toán*

## 2.4 Các ràng buộc

Ràng buộc khóa chính:

* production.categories: category\_id
* production.brands: brand\_id
* production.products: product\_id
* sales.customers: customer\_id
* sales.stores: store\_id
* sales.staffs: staff\_id, sales.orders: order\_id
* sales.order\_items: (order\_id, item\_id)
* production.stocks: (store\_id, product\_id)

Ràng buộc khóa ngoại:

* production.products: brand\_id references production.brands(brand\_id)
* production.products: category\_id references production.categories(category\_id)
* sales.staffs: store\_id references sales.stores(store\_id)
* sales.staffs: manager\_id references sales.staffs(staff\_id)
* sales.orders: customer\_id references sales.customers(customer\_id)
* sales.orders: store\_id references sales.stores(store\_id)
* sales.orders: staff\_id references sales.staffs(staff\_id)
* sales.order\_items: order\_id references sales.orders(order\_id)
* sales.order\_items: product\_id references production.products(product\_id)
* production.stocks: store\_id references sales.stores(store\_id)
* production.stocks: product\_id references production.products(product\_id)

Ràng buộc not null:

* production.categories: category\_name
* production.brands: brand\_name
* production.products: product\_name, brand\_id, category\_id, model\_year, list\_price
* sales.customers: first\_name, last\_name, email
* sales.stores: store\_name
* sales.staffs: first\_name, last\_name, email, active, store\_id
* sales.orders: order\_status, order\_date, required\_date, store\_id, staff\_id
* sales.order\_items: order\_id, product\_id, quantity, list\_price
* production.stocks: store\_id, product\_id, quantity

Các ràng buộc khác:

* sales.staffs: email là Unique
* sales.order\_items: (order\_id, item\_id) là Unique
* sales.order\_items: discount có giá trị mặc định là 0

**2.5 Bảng tầm ảnh hương của RBTV**

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| products - brands | + (brand\_id) | - (\*) | - |
| products - categories | + (category\_id) | - (\*) | - |
| staffs - stores | + (store\_id) | - (\*) | - |
| staffs - staffs (manager\_id) | + (manager\_id) | - (\*) | - |
| orders - customers | + (customer\_id) | - (\*) | - |
| orders - stores | + (store\_id) | - (\*) | - |
| orders - staffs | + (staff\_id) | - (\*) | - |
| order\_items - orders | + (order\_id) | - (\*) | - |
| order\_items - products | + (product\_id) | - (\*) | - |
| stocks - stores | + (store\_id) | - (\*) | - |
| stocks - products | + (product\_id) | - (\*) | - |

## 2.6 Tạo bảng và chèn dữ liệu

### 2.6.1 Tạo bảng:

Tạo schema để nhóm các bảng cùng loại:

CREATE SCHEMA production;

go

CREATE SCHEMA sales;

go

Bảng categories và brands:

CREATE TABLE production.categories (

category\_id INT IDENTITY (1, 1) PRIMARY KEY,

category\_name VARCHAR (255) NOT NULL

);

CREATE TABLE production.brands (

brand\_id INT IDENTITY (1, 1) PRIMARY KEY,

brand\_name VARCHAR (255) NOT NULL

);

Tạo bảng productions:

CREATE TABLE production.products (

product\_id INT IDENTITY (1, 1) PRIMARY KEY,

product\_name VARCHAR (255) NOT NULL,

brand\_id INT NOT NULL,

category\_id INT NOT NULL,

model\_year SMALLINT NOT NULL,

list\_price DECIMAL (10, 2) NOT NULL,

FOREIGN KEY (category\_id) REFERENCES production.categories (category\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (brand\_id) REFERENCES production.brands (brand\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

Tạọ bảng customer:

CREATE TABLE sales.customers (

customer\_id INT IDENTITY (1, 1) PRIMARY KEY,

first\_name VARCHAR (255) NOT NULL,

last\_name VARCHAR (255) NOT NULL,

phone VARCHAR (25),

email VARCHAR (255) NOT NULL,

street VARCHAR (255),

city VARCHAR (50),

state VARCHAR (25),

zip\_code VARCHAR (5)

);

Tạo bảng cửa hàng:

CREATE TABLE sales.stores (

store\_id INT IDENTITY (1, 1) PRIMARY KEY,

store\_name VARCHAR (255) NOT NULL,

phone VARCHAR (25),

email VARCHAR (255),

street VARCHAR (255),

city VARCHAR (255),

state VARCHAR (10),

zip\_code VARCHAR (5)

);

Tạo bảng nhân viên:

CREATE TABLE sales.staffs (

staff\_id INT IDENTITY (1, 1) PRIMARY KEY,

first\_name VARCHAR (50) NOT NULL,

last\_name VARCHAR (50) NOT NULL,

email VARCHAR (255) NOT NULL UNIQUE,

phone VARCHAR (25),

active tinyint NOT NULL,

store\_id INT NOT NULL,

manager\_id INT,

FOREIGN KEY (store\_id) REFERENCES sales.stores (store\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (manager\_id) REFERENCES sales.staffs (staff\_id) ON DELETE NO ACTION ON UPDATE NO ACTION

);

Tạo bảng orders:

CREATE TABLE sales.orders (

order\_id INT IDENTITY (1, 1) PRIMARY KEY,

customer\_id INT,

order\_status tinyint NOT NULL,

order\_date DATE NOT NULL,

required\_date DATE NOT NULL,

shipped\_date DATE,

store\_id INT NOT NULL,

staff\_id INT NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES sales.customers (customer\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (store\_id) REFERENCES sales.stores (store\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (staff\_id) REFERENCES sales.staffs (staff\_id) ON DELETE NO ACTION ON UPDATE NO ACTION

);

Tạo bảng order\_items tạo ra từ quan hệ giữa order và product:

CREATE TABLE sales.order\_items (

order\_id INT,

item\_id INT,

product\_id INT NOT NULL,

quantity INT NOT NULL,

list\_price DECIMAL (10, 2) NOT NULL,

discount DECIMAL (4, 2) NOT NULL DEFAULT 0,

PRIMARY KEY (order\_id, item\_id),

FOREIGN KEY (order\_id) REFERENCES sales.orders (order\_id) ON DELETE CASCADE ON UPDATE CASCADE,

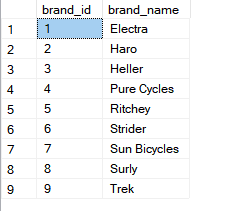
FOREIGN KEY (product\_id) REFERENCES production.products (product\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

2.6.2 Data

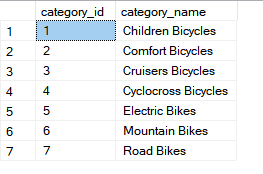
\*\* Vì dữ liệu có sẵn khá lớn, các dữ liệu được được lấy ra là mẫu nhỏ \*\*

--production.brands data:



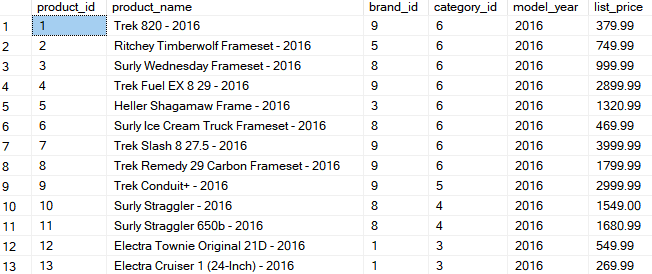
Hình 10. Bảng production.brands

--productions.categories data:



Hình 11. Bảng productions.categories

--productions.products data:



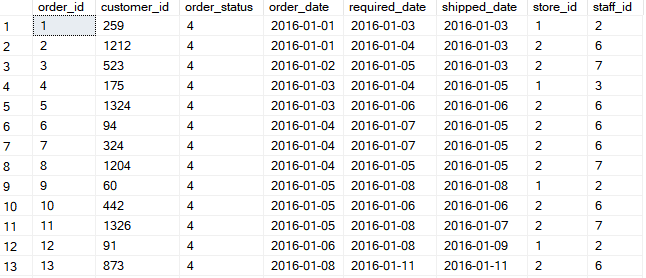
*Hình 12**. Bảng productions.products*

--productions.stocks data:



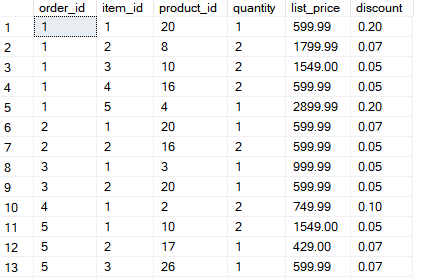
*Hình 13**. Bảng productions.stocks*

--sales.orders data:



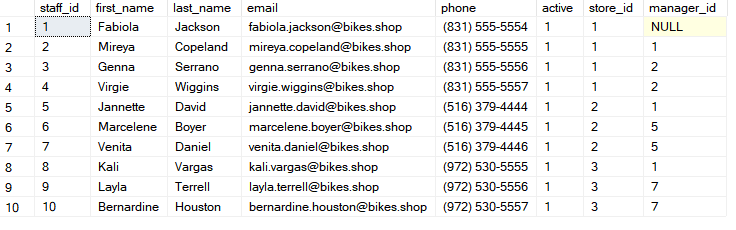
*Hình 14**. Bảng sales.orders*

--sales.order\_items data:



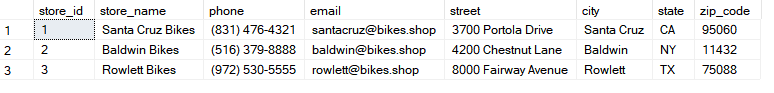
*Hình 15**. Bảng sales.order\_items*

--sales.staff data:



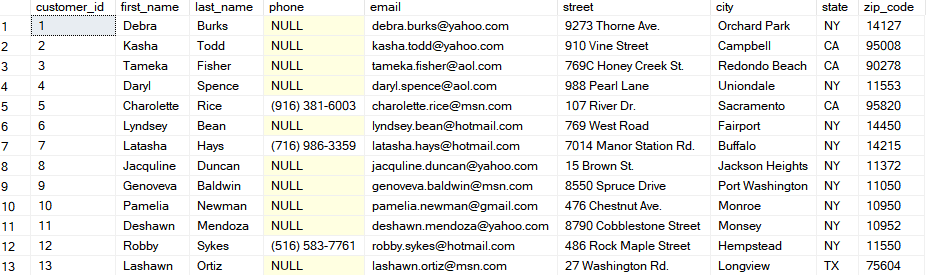
*Hình 16**. Bảng sales.staff*

--sales.stores data:



*Hình 17**. Bảng sales.stores*

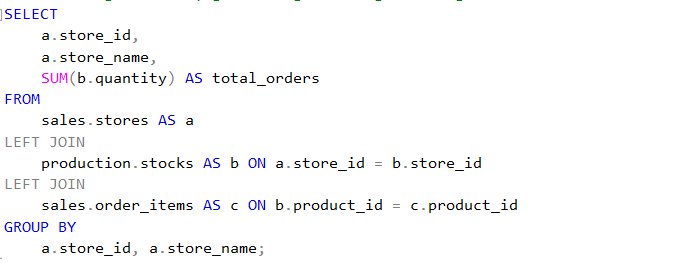
--sales.customers data:



*Hình 18**. Bảng sales.customers*

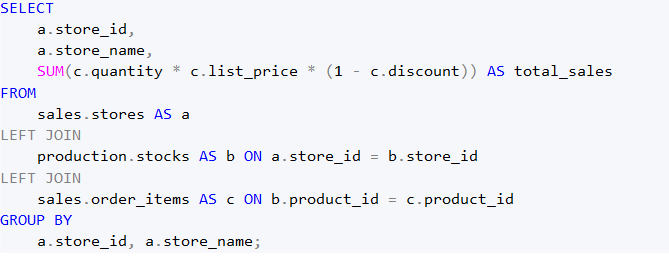
**2.7 Truy vấn dữ liệu**

-- 1. Thống kê số lượng đơn hàng của từng cửa hàng



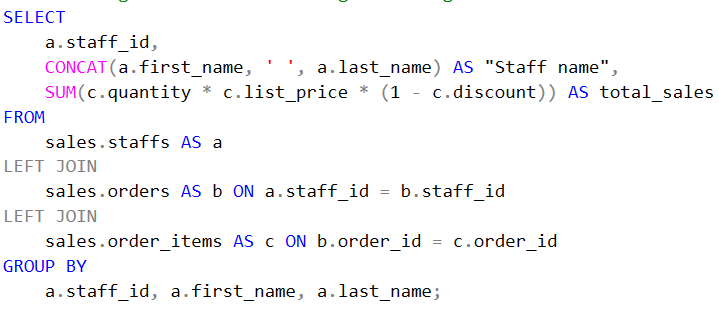


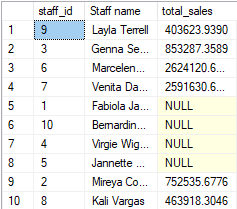
-- 2. Thống kê doanh số bán hàng của từng cửa hàng



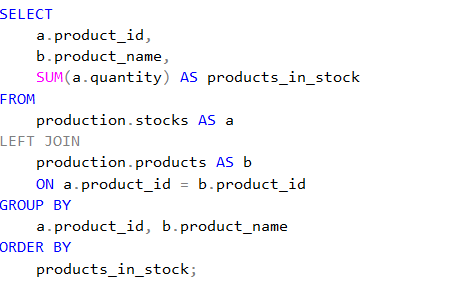


--3. Thống kê doanh số bán hàng theo từng nhân viên:



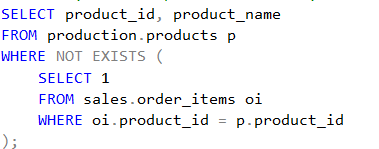


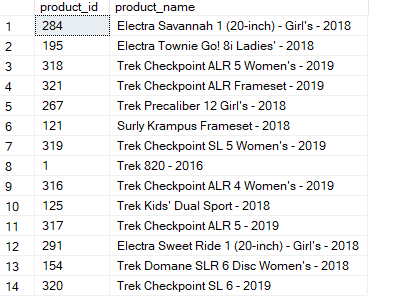
-- 4. Thống kê lượng tồn kho theo từng sản phẩm



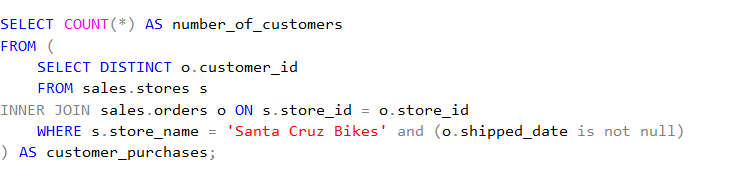


--5. liệt kê sản phẩm chưa bán được cái nào



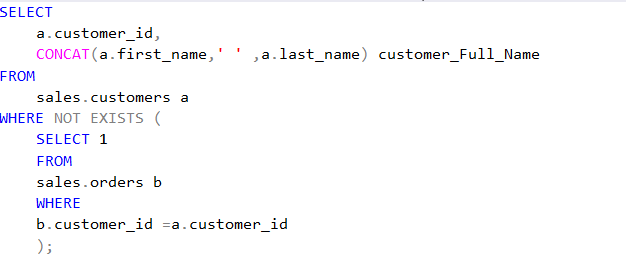


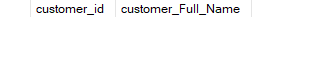
--6.Thống kê số khách hàng đã từng mua hàng tại cửa hàng Santa Cruz Bikes



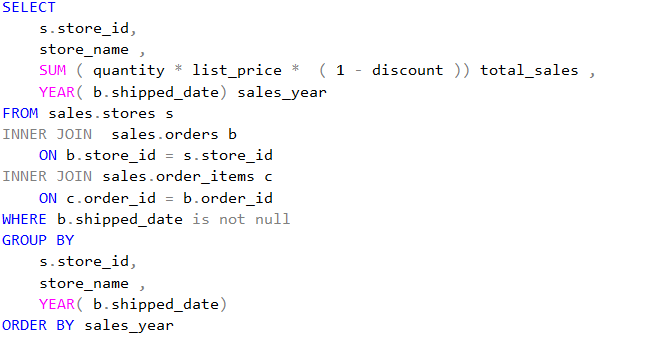


--7. liệt kê khách hàng chưa mua hàng lần nào





--8.thống kê doang số bán hàng của từng của hàng theo năm





# KẾT LUẬN

Bài toán về cơ sở dữ liệu hệ thống cơ sở dữ liệu cửa hàng xe đạp củng cố kiến thức về cách thức thiết kế cơ sở dữ liệu thông qua Entity model và chuyển đổi sang Relational model. Bài toán trên đã được chuẩn hóa nhằm tránh lặp lại dữ liệu và dễ truy xuất bằng cách tách các mỗi liện hệ thành các bảng riêng biệt. Ràng buộc giúp dữ liệu được thêm vào đúng và chuẩn theo bài toán đề ra. Sau 7 câu sql giúp cũng cố kiến thức về khai thác dữ liệu. Truy xuất đúng bài toán ta vần về dữ liệu đã được chèn vào bảng.

**TÀI LIỆU THAM KHẢO**

1. *Database System Concepts(2010)* by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
2. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling(2013)* by Ralph Kimball and Margy Ross
3. *<https://www.databasejournal.com/mysql>* Database Journal.