|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC ĐẠI NAM**  **KHOA CÔNG NGHỆ THÔNG TIN** | | | | | | | | | | | | | |
|  | | |  |  | |  | |  | |  | |  | |
|  | | |  | A logo with a black background  AI-generated content may be incorrect. | | | | | |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |  | |  | |  | |
| **BÀI TẬP LỚN**  **HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU** | | | | | | | | | | | | | |
|  | | |  |  | |  | |  | |  | |  | |
| **Đề tài:**  **THIẾT KẾ VÀ TRIỂN KHAI CƠ SỞ DỮ LIỆU CHO**  **HỆ THỐNG QUẢN LÝ XE BUÝT LIÊN TỈNH** | | | | | | | | | | | | | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | **Giảng viên hướng dẫn** | | | | | | : ThS. Nguyễn Ngọc Ân | | | | | |  |
|  | **Nhóm sinh viên thực hiện** | | | | | | : Nhóm 5 | | | | | |  |
|  |  | | | | | |  | | | | | |  |
|  | |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Họ và tên** | **Mã sinh viên** | **Lớp** | | 1 | Trần Minh Nghĩa | 1771020505 | CNTT 17-09 | | 2 | Trần Đức Lương | 1771020444 | CNTT 17-09 | | 3 | Đoàn Mai Lan | 1771020413 | CNTT 17-09 | | | | | | | | | | | | |  |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  | **Hà Nội, tháng 03 năm 2025** | | | | | |  | |  | |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC ĐẠI NAM**  **KHOA CÔNG NGHỆ THÔNG TIN** | | | | | | | | | | | | | |
|  | | |  |  | |  | |  | |  | |  | |
|  | | |  | A logo with a black background  AI-generated content may be incorrect. | | | | | |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |
|  | | |  |  | |  | |  | |  | |  | |
| **BÀI TẬP LỚN**  **HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU** | | | | | | | | | | | | | |
|  | | |  |  | |  | |  | |  | |  | |
| **Đề tài:**  **THIẾT KẾ VÀ TRIỂN KHAI CƠ SỞ DỮ LIỆU CHO**  **HỆ THỐNG QUẢN LÝ XE BUÝT LIÊN TỈNH** | | | | | | | | | | | | | |
|  | | |  |  | |  |  |  | |  | |  | |
| **STT** | | **Họ và tên** | | | **Mã sinh viên** | | **Ngày sinh** | | **Điểm** | | | | |
| **Bằng chữ** | | **Bằng số** | | |
| 1 | | Trần Minh Nghĩa | | | 1771020505 | | 02/08/2002 | |  | |  | | |
| 2 | | Trần Đức Lương | | | 1771020444 | | 21/06/2005 | |  | |  | | |
| 3 | | Đoàn Mai Lan | | | 1771020413 | | 22/10/2005 | |  | |  | | |
|  | | |  |  | |  |  |  | |  | |  | |
| **CÁN BỘ CHẤM THI 1** | | | | | | | **CÁN BỘ CHẤM THI 2** | | | | | | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
| **ThS. Nguyễn Ngọc Ân** | | | | | | |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  |  | |  |  |  | |  | |  | |
|  | | |  | **Hà Nội, tháng 03 năm 2025** | | | | | |  | |  | |

**MỤC LỤC**

[MỤC LỤC HÌNH ẢNH 6](#_Toc193295834)

[MỤC LỤC BẢNG 8](#_Toc193295835)

[MỤC LỤC KHUNG 9](#_Toc193295836)

[DANH MỤC TỪ VIẾT TẮT 11](#_Toc193295837)

[CHƯƠNG 1: 12](#_Toc193295838)

[GIỚI THIỆU ĐỀ TÀI 12](#_Toc193295839)

[1.1. Xây dựng bối cảnh giả định 12](#_Toc193295840)

[1.2. Mô tả bài toán 12](#_Toc193295841)

[1.3. Yêu cầu cần đạt được 13](#_Toc193295842)

[CHƯƠNG 2: 14](#_Toc193295843)

[PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 14](#_Toc193295844)

[2.1. Xây dựng mô hình thực thể quan hệ (Entity – Relationship Diagram – ERD) 14](#_Toc193295845)

[*2.1.1. Xác định các thực thể và các thuộc tính tương ứng* 14](#_Toc193295846)

[*2.1.2. Xác định liên kết và kiểu liên kết giữa các thực thể* 14](#_Toc193295847)

[*2.1.3. Xây dựng mô hình ERD* 16](#_Toc193295848)

[2.2. Xây dựng mô hình quan hệ 16](#_Toc193295849)

[*2.2.1. Các quy tắc chuyển hoá từ mô hình ERD sang mô hình quan hệ* 16](#_Toc193295850)

[*2.2.2. Chuyển hoá từ mô hình ERD sang mô hình quan hệ* 17](#_Toc193295851)

[2.3. Xác định các ràng buộc 17](#_Toc193295852)

[2.4. Xác định các phụ thuộc hàm và chuẩn hóa theo chuẩn 3NF 21](#_Toc193295853)

[*2.4.1. Chuẩn hóa bảng tbl.PHUONGTIEN* 21](#_Toc193295854)

[*2.4.2. Chuẩn hóa bảng tbl.TUYENXE* 21](#_Toc193295855)

[*2.4.3. Chuẩn hóa bảng tbl.TAIXE* 21](#_Toc193295856)

[*2.4.4. Chuẩn hóa bảng tbl.TRAMXE* 21](#_Toc193295857)

[*2.4.5. Chuẩn hóa bảng tbl.THANHPHO* 22](#_Toc193295858)

[*2.4.6. Chuẩn hóa bảng tbl.CHINHANH* 22](#_Toc193295859)

[CHƯƠNG 3: 23](#_Toc193295860)

[TRIỂN KHAI CƠ SỞ DỮ LIỆU 23](#_Toc193295861)

[3.1. Tạo cơ sở dữ liệu 23](#_Toc193295862)

[*3.1.1. Tạo cơ sở dữ liệu* 23](#_Toc193295863)

[*3.1.2. Tạo các bảng dữ liệu* 23](#_Toc193295864)

[*3.1.3. Cài đặt PK và FK* 24](#_Toc193295865)

[3.2. Nhập dữ liệu cho các bảng 24](#_Toc193295866)

[3.3. Truy vấn cơ bản (SELECT, INSERT, UPDATE, DELETE) 25](#_Toc193295867)

[*3.3.1. Truy vấn SELECT* 25](#_Toc193295868)

[*3.3.2. Truy vấn INSERT* 26](#_Toc193295869)

[*3.3.3. Truy vấn UPDATE* 27](#_Toc193295870)

[*3.3.4. Câu lệnh DELETE* 28](#_Toc193295871)

[3.4. Truy vấn nâng cao (INNER JOIN, GROUP BY, HAVING, SUBQUERY) 29](#_Toc193295872)

[*3.4.1. Truy vấn sử dụng INNER JOIN* 29](#_Toc193295873)

[*3.4.2. Truy vấn sử dụng GROUP BY* 30](#_Toc193295874)

[*3.4.3. Truy vấn sử dụng HAVING* 32](#_Toc193295875)

[*3.4.4. Truy vấn sử dụng SUBQUERY* 33](#_Toc193295876)

[CHƯƠNG 4: 37](#_Toc193295877)

[LẬP TRÌNH T-SQL 37](#_Toc193295878)

[4.1. Tạo view cho CSDL 37](#_Toc193295879)

[4.2. Tạo INDEX cần thiết cho các bảng 46](#_Toc193295880)

[4.3. Tạo STORED PROCEDURE 48](#_Toc193295881)

[*4.3.1. STORED PROCEDURE không tham số* 48](#_Toc193295882)

[*4.3.2. STORED PROCEDURE có tham số* 50](#_Toc193295883)

[*4.3.3. STORED PROCEDURE có đầu ra* 54](#_Toc193295884)

[4.4. Tạo các FUNCTION 56](#_Toc193295885)

[*4.4.1. FUNCTION trả về kiểu vô hướng* 56](#_Toc193295886)

[*4.4.2. FUNCTION trả về kiểu bảng* 58](#_Toc193295887)

[*4.4.3. FUNCTION sử dụng biến bảng* 60](#_Toc193295888)

[4.5. Tạo các TRIGGER để kiểm soát dữ liệu 62](#_Toc193295889)

[*4.5.1. TRIGGER theo dõi cập nhật, thay đổi dữ liệu các bảng* 62](#_Toc193295890)

[*4.5.2. Các TRIGGER khác* 65](#_Toc193295891)

[BẢO MẬT VÀ QUẢN TRỊ 69](#_Toc193295892)

[5.1. Tạo người dùng 69](#_Toc193295893)

[5.2. Phân cấp quyền truy cập của người dùng 70](#_Toc193295894)

[5.3. Quản lý sao lưu và phục hồi dữ liệu 71](#_Toc193295895)

[KẾT LUẬN 78](#_Toc193295896)

[PHÂN CÔNG NHIỆM VỤ 80](#_Toc193295897)

[ĐÁNH GIÁ NỘI BỘ 81](#_Toc193295898)

[TÀI LIỆU THAM KHẢO 82](#_Toc193295899)

# **MỤC LỤC HÌNH ẢNH**

[Ảnh 1. Mô hình ERD. 16](#_Toc193310372)

[Ảnh 2. Mô hình quan hệ. 17](#_Toc193310373)

[Ảnh 3. Kết quả thực hiện truy vấn SELECT. 26](#_Toc193310374)

[Ảnh 4. Số lượng bản ghi trong bảng TAIXE trước (25 bản ghi - ảnh trái) và sau (30 bản ghi - ảnh phải) khi thực hiện truy vấn INSERT. 27](#_Toc193310375)

[Ảnh 5. Dữ liệu của tuyến 01 và tài xế có mã 'TX6' trước (2 ảnh trên) và sau (2 ảnh dưới) khi thực hiện truy vấn UPDATE. 28](#_Toc193310376)

[Ảnh 6. Kết quả sau khi thực hiện truy xuất thông tin của phương tiện (ảnh trên), trạm dừng (ảnh dưới trái) và tài xế (ảnh dưới phải). 30](#_Toc193310377)

[Ảnh 7. Kết quả sau khi thực hiện truy vấn sử dụng GROUP BY. 32](#_Toc193310378)

[Ảnh 8. Kết quả sau khi thực hiện truy vấn sử dụng HAVING. 33](#_Toc193310379)

[Ảnh 9. Kết quả sau khi thực hiện truy vấn sử dụng SUBQUERY. 36](#_Toc193310380)

[Ảnh 10. Kết quả truy vấn VIEW theo dõi chi tiết phương tiện. 38](#_Toc193310381)

[Ảnh 11. Kết quả truy vấn VIEW chi tiết tài xế. 39](#_Toc193310382)

[Ảnh 12. Kết quả truy vấn VIEW chi tiết trạm xe. 40](#_Toc193310383)

[Ảnh 13. Kết quả truy vấn VIEW theo dõi tuổi tài xế. 41](#_Toc193310384)

[Ảnh 14. Kết quả thực hiện truy vấn VIEW theo dõi số năm còn có thể lao động của tài xế. 43](#_Toc193310385)

[Ảnh 15. Kết quả thực hiện truy vấn VIEW niên hạn của phương tiện. 44](#_Toc193310386)

[Ảnh 16. Kết quả sau khi thực hiện truy vấn VIEW theo dõi liên hệ của các chi nhánh. 45](#_Toc193310387)

[Ảnh 17. Kết quả sau khi thực hiện truy vấn VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý. 46](#_Toc193310388)

[Ảnh 18. Kết quả sau khi thực hiện truy vấn STORED PROCEDURE. 50](#_Toc193310389)

[Ảnh 19. Kết quả sau khi thực hiện STORED PROCEDURE có tham số. 54](#_Toc193310390)

[Ảnh 20. Kết quả sau khi thực hiện truy vấn STORED PROCEDURE có đầu ra. 56](#_Toc193310391)

[Ảnh 21. Kết quả sau khi thực hiện truy vấn FUNCTION trả về kiểu vô hướng. 58](#_Toc193310392)

[Ảnh 22. Kết quả sau khi thực hiện truy vấn FUNCTION trả về kiểu bảng. 60](#_Toc193310393)

[Ảnh 23. Kết quả sau khi thực hiện truy vấn FUNCTION sử dụng biến bảng. 62](#_Toc193310394)

[Ảnh 24. Kết quả sau khi cài đặt TRIGGER thành công, toàn bộ lịch sử thay đổi dữ liệu sẽ được lưu vết tại bảng LICHSUTHAYDOIDULIEU. 65](#_Toc193310395)

# **MỤC LỤC BẢNG**

[Bảng 1. Danh sách các ràng buộc. 17](#_Toc193294089)

[Bảng 2. Quy trình cài đặt backup tự động cho CSDL. 71](#_Toc193294090)

[Bảng 3. Quy trình backup thủ công cho thiết bị chạy hệ điều hành macOS. 76](#_Toc193294091)

[Bảng 4. Bảng phân công nhiệm vụ. 80](#_Toc193294092)

[Bảng 5. Bảng đánh giá thái độ và tinh thần tham gia. 81](#_Toc193294093)

[Bảng 6. Bảng đánh giá chất lượng công việc. 81](#_Toc193294094)

**MỤC LỤC KHUNG**

[Khung 1. Truy vấn khởi tạo cơ sở dữ liệu. 23](#_Toc193295978)

[Khung 2. Cấu trúc cơ bản của truy vấn khởi tạo bảng trong CSDL. 23](#_Toc193295979)

[Khung 3. Truy vấn khởi tạo bảng PHUONGTIEN. 23](#_Toc193295980)

[Khung 4. Cấu trúc truy vấn thêm PK và FK. 24](#_Toc193295981)

[Khung 5. Truy vấn thêm PK và FK cho bảng PHUONGTIEN. 24](#_Toc193295982)

[Khung 6. Cấu trúc cơ bản truy vấn nhập dữ liệu vào bảng. 24](#_Toc193295983)

[Khung 7. Truy vấn nhập dữ liệu vào bảng PHUONGTIEN. 25](#_Toc193295984)

[Khung 8. Truy vấn SELECT. 25](#_Toc193295985)

[Khung 9. Truy vấn thêm dữ liệu tài xế mới vào bảng TAIXE. 26](#_Toc193295986)

[Khung 10. Truy vấn UPDATE. 27](#_Toc193295987)

[Khung 11. Truy vấn DELETE. 29](#_Toc193295988)

[Khung 12. Truy vấn sử dụng INNER JOIN. 29](#_Toc193295989)

[Khung 13. Truy vấn sử dụng GROUP BY. 31](#_Toc193295990)

[Khung 14. Truy vấn sử dụng HAVING. 32](#_Toc193295991)

[Khung 15. Truy vấn sử dụng SUBQUERY. 34](#_Toc193295992)

[Khung 16. Khởi tạo VIEW theo dõi chi tiết phương tiện. 37](#_Toc193295993)

[Khung 17. Truy vấn VIEW theo dõi chi tiết phương tiện. 38](#_Toc193295994)

[Khung 18. Khởi tạo VIEW theo dõi chi tiết tài xế. 38](#_Toc193295995)

[Khung 19. Truy vấn VIEW chi tiết tài xế. 39](#_Toc193295996)

[Khung 20. Khởi tạo VIEW chi tiết trạm xe. 40](#_Toc193295997)

[Khung 21. Truy vấn VIEW chi tiết trạm xe. 40](#_Toc193295998)

[Khung 22. Khởi tạo VIEW theo dõi tuổi tài xế. 41](#_Toc193295999)

[Khung 23. Truy vấn VIEW theo dõi tuổi tài xế. 41](#_Toc193296000)

[Khung 24. Khởi tạo VIEW theo dõi số năm còn có thể lao động của tài xế. 41](#_Toc193296001)

[Khung 25. Truy vấn VIEW theo dõi số năm còn có thể lao động của tài xế. 42](#_Toc193296002)

[Khung 26. Khởi tạo VIEW theo dõi niên hạn của phương tiện. 43](#_Toc193296003)

[Khung 27. Truy vấn VIEW theo dõi niên hạn của phương tiện. 44](#_Toc193296004)

[Khung 28. Khởi tạo VIEW theo dõi dữ liệu liên hệ của các chi nhánh. 44](#_Toc193296005)

[Khung 29. Truy vấn VIEW theo dõi dữ liệu liên hệ của các chi nhánh. 45](#_Toc193296006)

[Khung 30. Khởi tạo VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý 45](#_Toc193296007)

[Khung 31. Truy vấn VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý. 46](#_Toc193296008)

[Khung 32. Khởi tạo chỉ mục (INDEX). 46](#_Toc193296009)

[Khung 33. Khởi tạo STORED PROCEDURE không tham số. 48](#_Toc193296010)

[Khung 34. Cấu trúc cơ bản truy vấn STORED PROCEDURE. 50](#_Toc193296011)

[Khung 35. Khởi tạo truy vấn STORED PROCEDURE có tham số. 50](#_Toc193296012)

[Khung 36. Truy vấn STORED PROCEDURE có tham số. 53](#_Toc193296013)

[Khung 37. Khởi tạo STORED PROCEDURE có đầu ra. 54](#_Toc193296014)

[Khung 38. Truy vấn STORED PROCEDURE có đầu ra. 55](#_Toc193296015)

[Khung 39. Khởi tạo FUNCTION trả về kiểu vô hướng 56](#_Toc193296016)

[Khung 40. Truy vấn FUNCTION trả về kiểu vô hướng. 57](#_Toc193296017)

[Khung 41. Khởi tạo FUNCTION trả về kiểu bảng. 58](#_Toc193296018)

[Khung 42. Truy vấn FUNCTION trả về kiểu bảng. 60](#_Toc193296019)

[Khung 43. Khởi tạo FUNCTION sử dụng biến bảng. 61](#_Toc193296020)

[Khung 44. Truy vấn FUNCTION sử dụng biến bảng. 62](#_Toc193296021)

[Khung 45. Tạo bảng lưu lịch sử thay đổi dữ liệu. 62](#_Toc193296022)

[Khung 46. Cài đặt TRIGGER cho các bảng (Ví dụ: Cài đặt cho bảng PHUONGTIEN). 63](#_Toc193296023)

[Khung 47. Khởi tạo TRIGGER ngăn xóa tài xế khi còn dữ liệu phương tiện liên quan. 65](#_Toc193296024)

[Khung 48. Khởi tạo TRIGGER kiểm tra biển kiểm soát của phương tiện. 66](#_Toc193296025)

[Khung 49. Khởi tạo TRIGGER ngăn chặn thay đổi thông tin mã tuyến. 66](#_Toc193296026)

[Khung 50. Khởi tạo TRIGGER chặn trùng lặp số điện thoại cho tài xế và chi nhánh. 67](#_Toc193296027)

[Khung 51. Khởi tạo TRIGGER kiểm tra số chỗ ngồi của phương tiện. 68](#_Toc193296028)

[Khung 52. Khởi tạo TRIGGER chặn trùng lặp vị trí của trạm xe. 69](#_Toc193296029)

[Khung 53. Khởi tạo người dùng trong CSDL. 69](#_Toc193296030)

[Khung 54. Cài đặt quyền cho từng đối tượng người dùng. 70](#_Toc193296031)

# **DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Từ đầy đủ** |
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | Transerco | Tổng công ty Vận tải Hà Nội |
| 3 | ERD | Entity – Relationship Diagram  (Mô hình thực thể - quan hệ) |
| 4 | PK | Primary Key (Khóa chính) |
| 5 | 1 - m | One to Many (Quan hệ một – nhiều) |
| 6 | 1 - 1 | One to One (Quan hệ một – một) |
| 7 | FK | Foreign Key (Khóa ngoại) |
| 8 | 3NF | Third Normal Form |
| 9 | 1NF | First Normal Form |
| 10 | 2NF | Second Normal Form |
| 11 | SQL MS20 | SQL Server Management Studio 20 |

# **CHƯƠNG 1:**

# **GIỚI THIỆU ĐỀ TÀI**

## **1.1. Xây dựng bối cảnh giả định**

Đặt tình huống bạn là chuyên viên quản trị cơ sở dữ liệu (CSDL) tại Tổng công ty Vận tải Hà Nội (Transerco). Bạn được giao nhiệm vụ thiết kế và triển khai xây dựng hệ CSDL cho hệ thống quản lý các tuyến xe cho tổng công ty.

## **1.2. Mô tả bài toán**

Tổng công ty Vận tải Hà Nội (Transerco) là Doanh nghiệp Nhà nước trực thuộc Ủy ban nhân dân thành phố Hà Nội, được thành lập từ năm 2004. Hiện nay, Transerco là đơn vị hàng đầu trong lĩnh vực cung cấp dịch vụ vận tải hành khách công cộng bằng xe buýt, quản lý bến bãi và điểm đỗ xe công cộng tại Thủ đô Hà Nội. Với hơn 10.000 lao động đang làm việc tại 12 đơn vị trực thuộc và 5 công ty con, để tối ưu hóa công tác quản lý, Tổng công ty có nhu cầu thiết kế một hệ CSDL đáp ứng các yêu cầu cụ thể sau:

Thứ nhất, hệ thống sẽ quản lý thông tin về các tuyến xe buýt do Transerco vận hành. Mỗi tuyến xe có một mã tuyến và tên tuyến riêng biệt với hành trình bắt đầu từ một trạm khởi hành và kết thúc tại một trạm kết thúc. Trên lộ trình, xe buýt có thể dừng đón – trả khách tại nhiều trạm trung gian khác nhau. Thông tin về các trạm dừng được lưu trữ bao gồm số trạm, tên trạm và vị trí địa lý của trạm đó. Một tuyến xe có thể đi qua nhiều trạm và ngược lại, một trạm có thể phục vụ nhiều tuyến xe khác nhau.

Thứ hai, bên cạnh việc quản lý các tuyến xe buýt và trạm dừng, hệ thống đồng thời đảm nhiệm việc quản lý các thành phố mà các tuyến xe buýt đi qua. Mỗi thành phố có một mã số thành phố và tên thành phố riêng. Một tuyến xe buýt có thể chạy qua nhiều thành phố (xe buýt đường dài) trên lộ trình của mình và một thành phố cũng có thể có nhiều tuyến xe buýt đi qua, tạo thành một mạng lưới vận tải liên kết giữa các khu vực.

Thứ ba, Transerco có nhiều chi nhánh hoạt động tại các địa phương khác nhau, mỗi chi nhánh có mã chi nhánh và tên chi nhánh riêng. Một chi nhánh được đặt tại một vị trí địa lý cụ thể, một thành phố có thể có nhiều chi nhánh cùng hoạt động. Vai trò của chi nhánh là quản lý các tuyến xe buýt trực thuộc, mỗi tuyến xe buýt sẽ thuộc quyền quản lý của 1 chi nhánh duy nhất. Việc xây dựng hệ thống chi nhánh giúp công ty phân vùng quản lý rõ ràng, từ đó tối ưu hoạt động điều hành và đảm bảo chất lượng dịch vụ.

Thứ tư, hệ thống cần theo dõi thông tin về các xe buýt hoạt động trong mạng lưới vận tải của tổng công ty. Mỗi xe buýt có một mã số xe, biển kiểm soát, số khung và số chỗ ngồi. Mỗi xe buýt được sử dụng cho một tuyến cụ thể, một tuyến có thể có nhiều xe buýt phục vụ.

Cuối cùng, hệ thống cũng cần có chức năng quản lý thông tin của các tài xế. Mỗi tài xế được quản lý dựa trên mã tài xế, họ tên, địa chỉ, số điện thoại và email. Việc phân công tài xế sẽ được thực hiện theo ngày, tức là mỗi xe buýt sẽ được giao cho một tài xế phụ trách trong một khoảng thời gian nhất định.

## **1.3. Yêu cầu cần đạt được**

Sau khi hoàn thành quá trình thiết kế và xây dựng, hệ thống CSDL quản lý xe buýt liên tỉnh của Transerco cần đạt được những yêu cầu sau:

- Đảm bảo lưu trữ đầy đủ, chính xác và đồng nhất tất cả các thông tin liên quan đến tuyến xe, trạm dừng, thành phố, chi nhánh, phương tiện và tài xế của tổng công ty. Dữ liệu phải được tổ chức một cách khoa học, tránh trùng lặp và đảm bảo dễ dàng cập nhật khi có sự thay đổi.

- Hỗ trợ phân công, điều phối tuyến xe một cách hiệu quả. Quản lý các tuyến xe từ điểm xuất phát đến điểm kết thúc, bao gồm danh sách các trạm dừng trên lộ trình. Đồng thời, hệ thống phải cho phép tra cứu thông tin một cách nhanh chóng, hỗ trợ tìm kiếm thông tin theo các tiêu chí như mã tuyến, tên tuyến, mã xe buýt, mã tài xế hoặc chi nhánh quản lý.

- Có cơ chế bảo mật và phân quyền, đảm bảo chỉ những người có thẩm quyền mới có thể truy cập hoặc chỉnh sửa thông tin.

- Đảm bảo hiệu suất hoạt động cao, khả năng mở rộng linh hoạt để có thể thêm tuyến xe mới, trạm dừng mới hoặc chi nhánh mới khi quy mô tổng công ty phát triển.

- Giao diện phải trực quan, dễ sử dụng, giúp nhân viên thao tác nhanh chóng và hạn chế sai sót.

# **CHƯƠNG 2:**

# **PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU**

## **2.1. Xây dựng mô hình thực thể quan hệ (Entity – Relationship Diagram – ERD)**

### ***2.1.1. Xác định các thực thể và các thuộc tính tương ứng***

a) Thực thể Tuyến\_Xe

- Mô tả: Lưu trữ thông tin về các tuyến xe.

- Thuộc tính: Mã tuyến (Primary Key – PK), tên tuyến, giờ bắt đầu chạy, giờ kết thúc, trạm khởi đầu, trạm trung gian và trạm kết thúc.

b) Thực thể Trạm\_Xe

- Mô tả: Lưu trữ thông tin về các trạm dừng xe.

- Thuộc tính: Mã trạm (PK) và vị trí trạm.

c) Thực thể Phương\_Tiện

- Mô tả: Lưu trữ thông tin về các phương tiện trong hệ thống.

- Thuộc tính: Mã phương tiện (PK), tên phương tiện, biển kiểm soát, tuyến phục vụ, hãng sản xuất, năm sản xuất, số khung và số chỗ ngồi.

d) Thực thể Tài\_Xế

- Mô tả: Lưu trữ thông tin về các tài xế.

- Thuộc tính: Mã tài xế, họ tên, ngày sinh, địa chỉ, phương tiện được giao quản lý, số điện thoại và email.

e) Thực thể Chi\_Nhánh

- Mô tả: Lưu trữ thông tin về các chi nhánh.

- Thuộc tính: Mã chi nhánh (PK), tên chi nhánh, địa chỉ, tuyến xe được giao quản lý, số điện thoại và email.

f) Thực thể Thành\_Phố

- Mô tả: Lưu trữ thông tin về các thành phố có mạng lưới xe buýt đi qua.

- Thuộc tính: Mã thành phố (PK), tên thành phố và tuyến xe chạy qua địa bàn,.

### ***2.1.2. Xác định liên kết và kiểu liên kết giữa các thực thể***

a) Quan hệ giữa Tuyến\_Xe và Phương\_Tiện

Mỗi tuyến xe sẽ có nhiều phương tiện phục vụ, mối quan hệ này là mối quan hệ một – nhiều (1 – m). Ngược lại, tại một thời điểm, một phương tiện chỉ có thể phục vụ một tuyến xe. Do đó, mối quan hệ này là mối quan hệ một – một (1 – 1).

b) Quan hệ giữa Tuyến\_Xe và Chi\_Nhánh

Mỗi tuyến xe do một chi nhánh quản lý, mối quan hệ này là mối quan hệ 1 – 1. Ngược lại, một chi nhánh có thể quản lý nhiều tuyến xe. Do đó, mối quan hệ này là mối quan hệ 1 – m.

c) Quan hệ giữa Tuyến\_Xe và Trạm\_Xe

Mỗi tuyến xe sẽ đi qua nhiều trạm xe trong lộ trình của mình, mối quan hệ này là mối quan hệ 1 – m. Tương tự, một trạm xe cũng sẽ có nhiều tuyến xe đi qua. Do đó, mối quan hệ này cũng là mối quan hệ 1 – m.

d) Quan hệ giữa Tuyến\_Xe và Thành\_Phố

Mỗi tuyến xe có thể đi qua nhiều thành phố khác nhau trong lộ trình của mình (xe buýt đường dài), mối quan hệ này là mối quan hệ 1 – m. Tương tự, một thành phố cũng sẽ có nhiều tuyến xe đi qua. Do đó, mối quan hệ này cũng là mối quan hệ 1 – m.

e) Quan hệ giữa Phương\_Tiện và Tài\_Xế

Mỗi phương tiện được giao cho một tài xế quản lý, mối quan hệ này là mối quan hệ 1 – 1. Tương tự, một tài xế cũng sẽ chỉ chịu trách nhiệm quản lý một phương tiện. Do đó, mối quan hệ này cũng là mối quan hệ 1 – 1.

f) Quan hệ giữa Tài\_Xế và Chi\_Nhánh

Mỗi tài xế sẽ do một chi nhánh quản lý, mối quan hệ này là mối quan hệ 1 – 1. Ngược lại, một chi nhánh có thể quản lý nhiều tài xế. Do đó, mối quan hệ này là mối quan hệ 1 – m.

g) Quan hệ giữa Chi\_Nhánh và Thành\_Phố

Mỗi chi nhánh sẽ nằm tại một thành phố, mối quan hệ này là mối quan hệ 1 – 1. Ngược lại, một thành phố có thể có nhiều chi nhánh. Do đó, mối quan hệ này là mối quan hệ 1 – m.

### ***2.1.3. Xây dựng mô hình ERD***

Từ những thực thể, thuộc tính của từng thực thể, liên kết và kiểu liên kết giữa các thực thể đã xác định tại *Mục 2.1.1.* và *Mục 2.1.2.* xây dựng được mô hình ERD như *Ảnh 1.* dưới:

A diagram of a flowchart

AI-generated content may be incorrect.

Ảnh 1. Mô hình ERD.

## **2.2. Xây dựng mô hình quan hệ**

### ***2.2.1. Các quy tắc chuyển hoá từ mô hình ERD sang mô hình quan hệ***

- Quy tắc 1: Mỗi thực thể trong mô hình ERD sẽ được chuyển thành một bảng trong mô hình quan hệ. Các thuộc tính của thực thể trở thành các cột của bảng đó. Nếu thực thể có PK, thực thể đó trở thành PK trong bảng tương ứng.

- Quy tắc 2: Đối với thuộc tính đa trị, tách thành một bảng riêng và đặt FK tham chiếu đến bảng đó vào bảng thực thể ban đầu.

- Quy tắc 3: Đối với các mối quan hệ 1 – 1, đặt PK của phía tuỳ chọn sang phía bắt buộc.

- Quy tắc 4: Đối với các mối quan hệ 1 – m, đặt PK của phía một sang phía nhiều.

- Quy tắc 5: Đối với các mối quan hệ m – m, tạo một bảng trung gian chứa PK của cả hai bảng thực thể ban đầu.

### ***2.2.2. Chuyển hoá từ mô hình ERD sang mô hình quan hệ***

Áp dụng các quy tắc chuyển hoá đã nêu ở *Mục 2.2.1.*, ta chuyển hoá mô hình ERD như *Ảnh 1.* thành mô hình quan hệ như *Ảnh 2.* dưới đây:

A diagram of a computer

AI-generated content may be incorrect.

Ảnh 2. Mô hình quan hệ.

## **2.3. Xác định các ràng buộc**

Bảng 1. Danh sách các ràng buộc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Bối cảnh** | **Nội dung**  **tự nhiên** | **Nội dung hình thức** |
| 1 | Ràng buộc 1 | tbl.PHUONGTIEN | Mỗi phương tiện chỉ do một và chỉ một tài xế quản lý | R1 = ∀ pt ∈ tbl.PHUONGTIEN: pt.mataixe ∈ tbl.TAIXE.mataixe |
| 2 | Ràng buộc 2 | tbl.PHUONGTIEN | Mỗi phương tiện chỉ hoạt động trong một và chỉ một tuyến xe | R2 = ∀ pt ∈ tbl.PHUONGTIEN: pt.matuyen ∈ tbl.TUYENXE.matuyen |
| 3 | Ràng buộc 3 | tbl.PHUONGTIEN | Mã phương tiện của mỗi phương tiện phải là duy nhất trong hệ thống | R3 = ∀ pt1, pt2 ∈ tbl.PHUONGTIEN: pt1 ≠ pt2 → pt1.maphuongtien ≠ pt2.maphuongtien |
| 4 | Ràng buộc 4 | tbl.PHUONGTIEN | Biển kiểm soát của mỗi phương tiện trên hệ thống không nhận giá trị rỗng | R4 = ∀ pt ∈ tbl.PHUONGTIEN: pt.bienkiemsoat ≠ NULL |
| 5 | Ràng buộc 5 | tbl.PHUONGTIEN | Số khung của mỗi phương tiện trên hệ thống không nhận giá trị rỗng | R5 = ∀ pt ∈ tbl.PHUONGTIEN: pt.sokhung ≠ NULL |
| 6 | Ràng buộc 6 | tbl.PHUONGTIEN | Năm sản xuất của các phương tiện trên hệ thống không nhận giá trị nhỏ hơn 2005 | R6 = ∀ pt ∈ tbl.PHUONGTIEN: pt.namsanxuat ≥ 2005 |
| 7 | Ràng buộc 7 | tbl.TUYENXE | Mỗi tuyến xe chỉ do một và chỉ một chi nhánh quản lý | R7 = ∀ tuyenxe ∈ tbl.TUYENXE: tuyenxe.matuyen ∈ tbl.CN\_TUYENXE.matuyen |
| 8 | Ràng buộc 8 | tbl.TUYENXE | Mã tuyến xe của mỗi tuyến xe phải là duy nhất trong hệ thống | R8 = ∀ tuyenxe1, tuyenxe2 ∈ tbl.TUYENXE: tuyenxe1 ≠ tuyenxe2 → tuyenxe1.maphuongtien ≠ tuyenxe2.maphuongtien |
| 9 | Ràng buộc 9 | tbl.TUYENXE | Giờ kết thúc của mỗi tuyến xe phải lớn hơn giờ bắt đầu của nó (Giờ được lấy theo định dạng 24 giờ/ngày) | R9 = ∀ tuyenxe ∈ tbl.TUYENXE: tuyenxe.gioketthuc > tuyenxe.giobatdau |
| 10 | Ràng buộc 10 | tbl.TAIXE | Mỗi tài xế chỉ do một và chỉ một chi nhánh quản lý | R10 = ∀ tx ∈ tbl.TAIXE: tx.machinhanh ∈ tbl.CHINHANH.machinhanh |
| 11 | Ràng buộc 11 | tbl.TAIXE | Mã tài xế của mỗi tài xế phải là duy nhất trong hệ thống | R11 = ∀ tx1, tx2 ∈ tbl.TAIXE: tx1 ≠ tx2 → tx1.mataixe ≠ tx2.mataixe |
| 12 | Ràng buộc 12 | tbl.TAIXE | Họ và tên của mỗi tài xế trên hệ thống không nhận giá trị rỗng | R12 = ∀ tx ∈ tbl.TAIXE: tx.hovaten ≠ NULL |
| 13 | Ràng buộc 13 | tbl.THANHPHO | Mã thành phố của mỗi thành phố phải là duy nhất trong hệ thống | R13 = ∀ tp1, tp2 ∈ tbl.THANHPHO: tp1 ≠ tp2 → tp1.mathanhpho ≠ tx2.mathanhpho |
| 14 | Ràng buộc 14 | tbl. THANHPHO | Tên thành phố của mỗi thành phố trên hệ thống không nhận giá trị rỗng | R14 = ∀ tp ∈ tbl.THANHPHO: tp.tenthanhpho ≠ NULL |
| 15 | Ràng buộc 15 | tbl.TRAMXE | Mã trạm xe của mỗi trạm xe phải là duy nhất trong hệ thống | R15 = ∀ txe1, txe2 ∈ tbl.TRAMXE: txe1 ≠ txe2 → txe1.matram ≠ txe2.matram |
| 16 | Ràng buộc 16 | tbl.TRAMXE | Vị trí trạm xe của mỗi trạm xe trên hệ thống không nhận giá trị rỗng | R16 = ∀ txe ∈ tbl.TRAMXE: txe.vitritram ≠ NULL |
| 17 | Ràng buộc 17 | tbl.CHINHANH | Mỗi chi nhánh chỉ thuộc một và chỉ một thành phố | R17 = ∀ cn ∈ tbl.CHINHANH: cn.mathanhpho ∈ tbl.THANHPHO.mathanhpho |
| 18 | Ràng buộc 18 | tbl.CHINHANH | Mã chi nhánh của mỗi chi nhánh phải là duy nhất trong hệ thống | R18 = ∀ cn1, cn2 ∈ tbl.CHINHANH: cn1 ≠ cn2 → cn1.machinhanh ≠ cn2.machinhanh |
| 19 | Ràng buộc 19 | tbl.CHINHANH | Tên chi nhánh của mỗi chi nhánh trên hệ thống không nhận giá trị rỗng | R19 = ∀ cn ∈ tbl.CHINHANH: cn.tenchinhanh ≠ NULL |
| 29 | Ràng buộc 20 | tbl.CHINHANH | Địa chỉ chi nhánh của mỗi chi nhánh trên hệ thống không nhận giá trị rỗng | R20 = ∀ cn ∈ tbl.CHINHANH: cn.diachi ≠ NULL |

## **2.4. Xác định các phụ thuộc hàm và chuẩn hóa theo chuẩn 3NF**

### ***2.4.1. Chuẩn hóa bảng tbl.PHUONGTIEN***

tbl.PHUONGTIEN (Mã phương tiện (PK), Tên phương tiện, Biển kiểm soát, Hãng sản xuất, Năm sản xuất, Số khung, Số chỗ ngồi, Mã tài xế (FK), Mã tuyến (FK)).

- Không có thuộc tính nào trong bảng tbl.PHUONGTIEN là thuộc tính đa trị. Do đó, bảng tbl.PHUONGTIEN đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.PHUONGTIEN đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.PHUONGTIEN đạt chuẩn 2NF và 3NF.

### ***2.4.2. Chuẩn hóa bảng tbl.TUYENXE***

tbl.TUYENXE (Mã tuyến (PK), Tên tuyến, Giờ bắt đầu chạy, Giờ kết thúc, Mã chi nhánh).

- Không có thuộc tính nào trong bảng tbl.TUYENXE là thuộc tính đa trị. Do đó, bảng tbl.TUYENXE đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.TUYENXE đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.TUYENXE đạt chuẩn 2NF và 3NF.

### ***2.4.3. Chuẩn hóa bảng tbl.TAIXE***

tbl.TAIXE (Mã tài xế (PK), Họ và tên, Ngày sinh, Địa chỉ, Mã chi nhánh (FK)).

- Không có thuộc tính nào trong bảng tbl.TAIXE là thuộc tính đa trị. Do đó, bảng tbl.TAIXE đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.TAIXE đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.TAIXE đạt chuẩn 2NF và 3NF.

### ***2.4.4. Chuẩn hóa bảng tbl.TRAMXE***

tbl.TRAMXE (Mã trạm (PK), Vị trí trạm).

- Không có thuộc tính nào trong bảng tbl.TRAMXE là thuộc tính đa trị. Do đó, bảng tbl.TRAMXE đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.TRAMXE đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.TRAMXE đạt chuẩn 2NF và 3NF.

### ***2.4.5. Chuẩn hóa bảng tbl.THANHPHO***

tbl.THANHPHO (Mã thành phố (PK), Tên thành phố).

- Không có thuộc tính nào trong bảng tbl.THANHPHO là thuộc tính đa trị. Do đó, bảng tbl.THANHPHO đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.THANHPHO đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.THANHPHO đạt chuẩn 2NF và 3NF.

### ***2.4.6. Chuẩn hóa bảng tbl.CHINHANH***

tbl.CHINHANH (Mã chi nhánh (PK), Tên chi nhánh, Địa chỉ, Mã thành phố (FK)).

- Không có thuộc tính nào trong bảng tbl.CHINHANH là thuộc tính đa trị. Do đó, bảng tbl.CHINHANH đạt chuẩn 1NF.

- Tất cả các thuộc tính không khóa trong bảng tbl.CHINHANH đều phụ thuộc vào PK, không có phụ thuộc hàm một phần và không phụ thuộc vào bất kì thuộc tính không khóa nào khác. Do đó, bảng tbl.CHINHANH đạt chuẩn 2NF và 3NF.

# **CHƯƠNG 3:**

# **TRIỂN KHAI CƠ SỞ DỮ LIỆU**

## **3.1. Tạo cơ sở dữ liệu**

### ***3.1.1. Tạo cơ sở dữ liệu***

Khung 1. Truy vấn khởi tạo cơ sở dữ liệu.

|  |
| --- |
| CREATE DATABASE Transerco\_Bus\_Management\_DB |

### ***3.1.2. Tạo các bảng dữ liệu***

Khung 2. Cấu trúc cơ bản của truy vấn khởi tạo bảng trong CSDL.

|  |
| --- |
| CREATE TABLE tên\_bảng(  tên\_cột\_1 kiểu\_dữ\_liệu\_1 [RÀNG\_BUỘC\_NẾU\_CÓ],  tên\_cột\_2 kiểu\_dữ\_liệu\_2 [RÀNG\_BUỘC\_NẾU\_CÓ],  ...  tên\_cột\_n kiểu\_dữ\_liệu\_n [RÀNG\_BUỘC\_NẾU\_CÓ],  ); |

Ví dụ 1: Truy vấn khởi tạo bảng PHUONGTIEN.

Khung 3. Truy vấn khởi tạo bảng PHUONGTIEN.

|  |
| --- |
| CREATE TABLE PHUONGTIEN(  Ma\_Phuong\_Tien NVARCHAR(10),  Ma\_Tuyen NVARCHAR(10),  Ma\_Tai\_Xe NVARCHAR(10),  Ten\_Phuong\_Tien NVARCHAR(50),  Bien\_Kiem\_Soat NVARCHAR(12),  Hang\_San\_Xuat NVARCHAR (50),  Nam\_San\_Xuat INT,  So\_Khung VARCHAR(17),  So\_Cho\_Ngoi INT  ); |

### ***3.1.3. Cài đặt PK và FK***

Khung 4. Cấu trúc truy vấn thêm PK và FK.

|  |
| --- |
| -- Cấu trúc truy vấn thêm khóa chính  ALTER TABLE tên\_bảng ADD CONSTRAINT tên\_ràng\_buộc PRIMARY KEY (tên\_cột);  -- Cấu trúc truy vấn thêm khóa ngoại  ALTER TABLE tên\_bảng\_con  ADD CONSTRAINT tên\_ràng\_buộc  FOREIGN KEY (tên\_cột\_con)  REFERENCES tên\_bảng\_cha (tên\_cột\_cha); |

Ví dụ 2: Truy vấn thêm PK và FK cho bảng PHUONGTIEN.

Khung 5. Truy vấn thêm PK và FK cho bảng PHUONGTIEN.

|  |
| --- |
| -- Thêm PK  ALTER TABLE PHUONGTIEN ADD CONSTRAINT PK\_PHUONGTIEN PRIMARY KEY (Ma\_Phuong\_Tien);  -- Thêm FK  ALTER TABLE PHUONGTIEN ADD CONSTRAINT FK\_PHUONGTIEN\_TUYENXE FOREIGN KEY (Ma\_Tuyen) REFERENCES TUYENXE (Ma\_Tuyen);  ALTER TABLE PHUONGTIEN ADD CONSTRAINT FK\_PHUONGTIEN\_TAIXE FOREIGN KEY (Ma\_Tai\_Xe) REFERENCES TAIXE (Ma\_Tai\_Xe); |

## **3.2. Nhập dữ liệu cho các bảng**

Khung 6. Cấu trúc cơ bản truy vấn nhập dữ liệu vào bảng.

|  |
| --- |
| INSERT INTO tên\_bảng (cột\_1, cột\_2, cột\_3, ...)  VALUES (giá\_trị\_1, giá\_trị\_2, giá\_trị\_3, ...); |

Ví dụ 3: Truy vấn nhập dữ liệu vào bảng PHUONGTIEN.

Khung 7. Truy vấn nhập dữ liệu vào bảng PHUONGTIEN.

|  |
| --- |
| INSERT INTO PHUONGTIEN (Ma\_Phuong\_Tien, Ma\_Tuyen, Ma\_Tai\_Xe, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat, Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi)  VALUES  ('VH1', 'BUS1', 'TX1', 'Samco City D60', '29B - 184.23', 'Samco', '2020', 'RL4BB42A0R1234567', '60'),  ('VH2', 'BUS1', 'TX2', 'Samco City D60', '29B - 937.58', 'Samco', '2020', 'RL4BC32F8R2345678', '60'),  ('VH3', 'BUS1', 'TX3', 'Samco City D60', '29B - 462.91', 'Samco', '2020', 'RL4BD21G9R3456789', '60'),  ('VH4', 'BUS1', 'TX4', 'Samco City D60', '29B - 715.36', 'Samco', '2020', 'RL4BE51H1R4567890', '60'),  ('VH5', 'BUS1', 'TX5', 'Samco City D60', '29B - 528.47', 'Samco', '2020', 'RL4BF62J2R5678901', '60'),  ...  ('VH25', 'BUS11', 'TX25', 'Thaco Meadow 89CT', '29B - 967.53', 'Thaco', '2024', 'RL4CB82E4R5678901', '60'); |

## **3.3. Truy vấn cơ bản (SELECT, INSERT, UPDATE, DELETE)**

### ***3.3.1. Truy vấn SELECT***

Ví dụ 4: Thực hiện truy vấn theo các yêu cầu sau:

a) Lấy toàn bộ thông tin trong bảng PHUONGTIEN.

b) Lấy thông tin phương tiện có mã phương tiện ‘VH25’.

c) Lấy thông tin toàn bộ phương tiện do hãng Samco sản xuất năm 2023 và có 50 chỗ ngồi.

Khung 8. Truy vấn SELECT.

|  |
| --- |
| -- a) Lấy toàn bộ dữ liệu trong bảng phương tiện  SELECT \* FROM PHUONGTIEN;  -- b) Lấy dữ liệu phương tiện có mã 'VH25'  SELECT \* FROM PHUONGTIEN WHERE Ma\_Phuong\_Tien = 'VH25';  -- c) Lấy toàn bộ dữ liệu các phương tiện do Samco sản xuất -- năm 2023 có số chỗ ngồi là 50  SELECT \* FROM PHUONGTIEN WHERE  Hang\_San\_Xuat = 'Samco' AND Nam\_San\_Xuat = 2023 AND So\_Cho\_Ngoi = 50; |

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. | |
| A screenshot of a computer  AI-generated content may be incorrect. | A screenshot of a computer  AI-generated content may be incorrect. |

Ảnh 3. Kết quả thực hiện truy vấn SELECT.

### ***3.3.2. Truy vấn INSERT***

Ví dụ 5: Thêm dữ liệu của 5 tài xế mới có mã từ ‘TX26’ đến ‘TX30’ vào bảng TAIXE.

Khung 9. Truy vấn thêm dữ liệu tài xế mới vào bảng TAIXE.

|  |
| --- |
| INSERT INTO TAIXE (Ma\_Tai\_Xe, Ma\_Chi\_Nhanh, Ho\_Va\_Ten, Ngay\_Sinh, Dia\_Chi)  VALUES  ('TX26', 'CN8', N'Ngô Duy Hưng', '1970-03-19', N'Số 112, đường Lê Trọng Tấn, quận Thanh Xuân, Hà Nội'),  ('TX27', 'CN8', N'Nguyễn Thành Long', '1973-02-01', N'Số 435, đường Xuân Thủy, quận Cầu Giấy, Hà Nội'),  ('TX28', 'CN8', N'Phạm Minh Quang', '1979-01-10', N'Số 213, phố Láng Hạ, quận Đống Đa, Hà Nội'),  ('TX29', 'CN8', N'Đào Mạnh Hùng', '1972-11-18', N'Số 69, đường Đại Cồ Việt, quận Hai Bà Trưng, Hà Nội'),  ('TX30', 'CN8', N'Lưu Thành Quân', '1972-08-25', N'Số 189, đường Nguyễn Chí Thanh, quận Ba Đình, Hà Nội'); |

|  |  |
| --- | --- |
|  |  |

Ảnh 4. Số lượng bản ghi trong bảng TAIXE trước (25 bản ghi - ảnh trái) và sau (30 bản ghi - ảnh phải) khi thực hiện truy vấn INSERT.

### ***3.3.3. Truy vấn UPDATE***

Ví dụ 6: Sử dụng truy vấn UPDATE để thực hiện các yêu cầu sau:

a) Cập nhật thông tin giờ kết thúc cho tuyến xe 01 trong bảng TUYENXE thành 22:00.

b) Cập nhật số điện thoại mới cho tài xế có mã tài xế ‘TX6’ trong bảng TAIXE\_LIENHE.

Khung 10. Truy vấn UPDATE.

|  |
| --- |
| -- a) Thay đổi giờ kết thúc tuyến 01  -- Kiểm tra thông tin tuyến 01  SELECT \* FROM TUYENXE WHERE Ma\_Tuyen = 'BUS1';  -- Cập nhật giờ kết thúc mới  UPDATE TUYENXE SET Gio\_Ket\_Thuc = '22:00' WHERE Ma\_Tuyen = 'BUS1'  -- b) Thay đổi số điện thoại tài xế có mã tài xế 'TX6'  -- Kiểm tra thông tin tài xế có mã tài xế 'TX6'  SELECT \* FROM TAIXE\_LIENHE  WHERE Ma\_Tai\_Xe = 'TX6' AND Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại';  -- Cập nhật số điện thoại mới  UPDATE TAIXE\_LIENHE SET Gia\_Tri = '0923 738 321'  WHERE Ma\_Tai\_Xe = 'TX6' AND Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại'; |

|  |  |
| --- | --- |
|  | A screenshot of a computer  AI-generated content may be incorrect. |
| A screenshot of a computer  AI-generated content may be incorrect. | A screenshot of a computer  AI-generated content may be incorrect. |

Ảnh 5. Dữ liệu của tuyến 01 và tài xế có mã 'TX6' trước (2 ảnh trên) và sau (2 ảnh dưới) khi thực hiện truy vấn UPDATE.

### ***3.3.4. Câu lệnh DELETE***

Ví dụ 7: Sử dụng truy vấn DELETE xóa các tài xế có mã từ ‘TX26’ tới ‘TX30’ trong bảng TAIXE.

Khung 11. Truy vấn DELETE.

|  |
| --- |
| DELETE FROM TAIXE WHERE Ma\_Tai\_XE  IN ('TX26', 'TX27', 'TX28', 'TX29', 'TX30'); |

## **3.4. Truy vấn nâng cao (INNER JOIN, GROUP BY, HAVING, SUBQUERY)**

### ***3.4.1. Truy vấn sử dụng INNER JOIN***

Ví dụ 8: Thực hiện các yêu cầu sau bằng cách sử dụng INNER JOIN.

a) Lấy dữ liệu phương tiện và tài xế điều khiển phương tiện đó.

b) Lấy dữ liệu vị trí các điểm dừng của tuyến 01.

c) Lấy tất cả dữ liệu của tài xế có mã ‘TX10’ trong CSDL.

Khung 12. Truy vấn sử dụng INNER JOIN.

|  |
| --- |
| -- a) Lấy dữ liệu phương tiện và tài xế điều khiển chúng  SELECT  pt.Ma\_Phuong\_Tien, pt.Ten\_Phuong\_Tien, pt.Bien\_Kiem\_Soat, pt.Ma\_Tai\_Xe,  tx.Ho\_Va\_Ten  FROM PHUONGTIEN pt INNER JOIN TAIXE tx ON pt.Ma\_Tai\_Xe = tx.Ma\_Tai\_Xe;  -- b) Lấy dữ liệu các điểm dừng của tuyến 01  SELECT  lt.Ma\_Tuyen, lt.Ma\_Tram,  trx.Vi\_Tri\_Tram,  lt.Thu\_Tu\_Tram  FROM LOTRINH lt INNER JOIN TRAMXE trx  ON lt.Ma\_Tram = trx.Ma\_Tram WHERE lt.Ma\_Tuyen = 'BUS1';  -- c) Lấy tất cả dữ liệu của tài xế có mã tài xế 'TX10'  SELECT  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten, tx.Ngay\_Sinh, tx.Dia\_Chi,  MAX(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN txlh.Gia\_Tri END) AS Email,  MAX(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN txlh.Gia\_Tri END) AS So\_Dien\_Thoai  FROM TAIXE tx INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  WHERE tx.Ma\_Tai\_Xe = 'TX10'  GROUP BY tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten, tx.Ngay\_Sinh, tx.Dia\_Chi; |

|  |  |
| --- | --- |
|  | |
|  |  |

Ảnh 6. Kết quả sau khi thực hiện truy xuất thông tin của phương tiện (ảnh trên), trạm dừng (ảnh dưới trái) và tài xế (ảnh dưới phải).

### ***3.4.2. Truy vấn sử dụng GROUP BY***

Ví dụ 9: Thực hiện các yêu cầu sau bằng cách sử dụng GROUP BY.

a) Đếm số tài xế của mỗi chi nhánh.

b) Đếm số trạm dừng của mỗi tuyến.

c) Đếm số trạm dừng trên lộ trình của tuyến có mã ‘BUS8’.

d) Đếm số phương tiện do Thaco sản xuất.

Khung 13. Truy vấn sử dụng GROUP BY.

|  |
| --- |
| -- a) Đếm số tài xế của mỗi chi nhánh  SELECT Ma\_Chi\_Nhanh, COUNT(\*) AS So\_Tai\_Xe  FROM TAIXE  GROUP BY Ma\_Chi\_Nhanh;  -- b) Đếm số trạm dừng của mỗi tuyến  SELECT Ma\_Tuyen, COUNT(\*) AS So\_Tram\_Dung  FROM LOTRINH  GROUP BY Ma\_Tuyen;  -- c) Đếm số trạm dừng trên lộ trình tuyến buýt có mã 'BUS8'  SELECT Ma\_Tuyen, COUNT(\*) AS So\_Tram\_Dung  FROM LOTRINH  WHERE Ma\_Tuyen = 'BUS8'  GROUP BY Ma\_Tuyen;  -- d) Đếm số phương tiện do Thaco  SELECT Hang\_San\_Xuat, COUNT(\*) AS So\_Luong  FROM PHUONGTIEN  WHERE Hang\_San\_Xuat = 'Thaco'  GROUP BY Hang\_San\_Xuat; |

|  |  |
| --- | --- |
|  |  |
|  |  |

Ảnh 7. Kết quả sau khi thực hiện truy vấn sử dụng GROUP BY.

### ***3.4.3. Truy vấn sử dụng HAVING***

Ví dụ 10: Thực hiện các yêu cầu sau bằng cách sử dụng HAVING.

a) Lọc ra các chi nhánh có nhiều hơn 6 tài xế.

b) Lọc các tuyến xe có nhiều hơn 30 trạm dừng.

c) Lọc các chi nhánh quản lý nhiều hơn một tuyến xe.

Khung 14. Truy vấn sử dụng HAVING.

|  |
| --- |
| -- a) Lọc chi nhánh có nhiều hơn 6 tài xế  SELECT  tx.Ma\_Chi\_Nhanh,  cn.Ten\_Chi\_Nhanh, cn.Dia\_Chi,  COUNT(\*) AS So\_Luong  FROM TAIXE tx INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = cn.Ma\_Chi\_Nhanh  GROUP BY tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh, cn.Dia\_Chi  HAVING COUNT(\*) > 6;  -- b) Lọc các tuyến xe có nhiều hơn 30 trạm dừng  SELECT  lt.Ma\_Tuyen,  tx.Ten\_Tuyen,  COUNT(\*) AS So\_Tram  FROM LOTRINH lt INNER JOIN TUYENXE tx ON lt.Ma\_Tuyen = tx.Ma\_Tuyen  GROUP BY lt.Ma\_Tuyen, tx.Ten\_Tuyen  HAVING COUNT(\*) > 30;  -- c) Lọc các chi nhánh quản lý nhiều hơn 1 tuyến xe  SELECT  cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  COUNT(\*) AS So\_Tuyen,  cn.Dia\_Chi  FROM CHINHANH cn INNER JOIN TUYENXE tx ON cn.Ma\_Chi\_Nhanh = tx.Ma\_Chi\_Nhanh  GROUP BY cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh, tx.Ma\_Chi\_Nhanh, cn.Dia\_Chi  HAVING COUNT(\*) > 1; |

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. | |
| A screenshot of a computer  AI-generated content may be incorrect. | A screenshot of a computer  AI-generated content may be incorrect. |

Ảnh 8. Kết quả sau khi thực hiện truy vấn sử dụng HAVING.

### ***3.4.4. Truy vấn sử dụng SUBQUERY***

Ví dụ 11: Thực hiện các yêu cầu sau bằng cách sử dụng SUBQUERY.

a) Lấy dữ liệu các tuyến xe thuộc chi nhánh 1 mà có số điểm dừng lớn hơn 20.

b) Lấy dữ liệu các trạm xe có nhiều hơn 1 tuyến xe đi qua.

c) Lấy dữ liệu tài xế và các phương tiện có nhiều hơn 40 chỗ ngồi thuộc quản lý của chi nhánh 6.

d) Lấy dữ liệu các phương tiện có giờ kết thúc là 21:00.

Khung 15. Truy vấn sử dụng SUBQUERY.

|  |
| --- |
| -- a) Lấy dữ liệu các tuyến xe thuộc chi nhánh 1 mà có số  -- điểm dừng lớn hơn 20  SELECT  tx.Ma\_Tuyen, tx.Ten\_Tuyen,  cn.Ten\_Chi\_Nhanh,  lt.So\_Tram  FROM TUYENXE tx  INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = cn.Ma\_Chi\_Nhanh  INNER JOIN  (SELECT Ma\_Tuyen, COUNT(\*) AS So\_Tram FROM LOTRINH  GROUP BY Ma\_Tuyen HAVING COUNT(\*) > 20) lt ON tx.Ma\_Tuyen = lt.Ma\_Tuyen  WHERE tx.Ma\_Chi\_Nhanh = 'CN1';  -- b) Lấy dữ liệu các trạm có nhiều hơn 1 tuyến xe đi qua  SELECT  tx.Ma\_Tram, tx.Vi\_Tri\_Tram,  lt.So\_Tuyen  FROM TRAMXE tx  INNER JOIN  (SELECT Ma\_Tram, COUNT(\*) AS So\_Tuyen FROM LOTRINH  GROUP BY Ma\_Tram HAVING COUNT(\*) > 1) lt ON tx.Ma\_Tram = lt.Ma\_Tram;  -- c) Lấy dữ liệu tài xế và các phương tiện có số chỗ ngồi  -- lớn hơn 40 thuộc quản lý của chi nhánh 6  SELECT  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten, tx.Ma\_Chi\_Nhanh, tx.Ngay\_Sinh, tx.Dia\_Chi,  pt.Ma\_Phuong\_Tien, pt.Ten\_Phuong\_Tien, pt.Bien\_Kiem\_Soat, pt.So\_Cho\_Ngoi  FROM TAIXE tx  INNER JOIN  (SELECT Ma\_Tai\_Xe, Ma\_Phuong\_Tien,  Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, So\_Cho\_Ngoi  FROM PHUONGTIEN WHERE So\_Cho\_Ngoi > 40) pt ON tx.Ma\_Tai\_Xe = pt.Ma\_Tai\_Xe  WHERE tx.Ma\_Chi\_Nhanh = 'CN6';  -- d) Lấy dữ liệu các phương tiện có giờ kết thúc là 21:00  SELECT  pt.Ma\_Phuong\_Tien, pt.Ten\_Phuong\_Tien, pt.Bien\_Kiem\_Soat,  tx.Gio\_Ket\_Thuc  FROM PHUONGTIEN pt  INNER JOIN  (SELECT Ma\_Tuyen, Gio\_Ket\_Thuc  FROM TUYENXE WHERE Gio\_Ket\_Thuc = '21:00')  tx ON pt.Ma\_Tuyen = tx.Ma\_Tuyen; |

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. |  |
|  |  |

Ảnh 9. Kết quả sau khi thực hiện truy vấn sử dụng SUBQUERY.

# **CHƯƠNG 4:**

# **LẬP TRÌNH T-SQL**

## **4.1. Tạo view cho CSDL**

Ví dụ 12: Tạo VIEW để theo dõi tài xế, tuyến và chi nhánh quản lý của 1 phương tiện.

Khung 16. Khởi tạo VIEW theo dõi chi tiết phương tiện.

|  |
| --- |
| GO  CREATE VIEW view\_CHITIET\_PHUONGTIEN AS  SELECT  pt.Ma\_Phuong\_Tien,  pt.Ma\_Tuyen, tx.Ten\_Tuyen,  tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  pt.Ma\_Tai\_Xe, txe.Ho\_Va\_Ten,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN txlh.Gia\_Tri END, ', ') AS Email,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN txlh.Gia\_Tri END, ', ') AS So\_Dien\_Thoai,  pt.Ten\_Phuong\_Tien, pt.Bien\_Kiem\_Soat, pt.Hang\_San\_Xuat, pt.Nam\_San\_Xuat,  pt.So\_Khung, pt.So\_Cho\_Ngoi  FROM PHUONGTIEN pt  INNER JOIN TUYENXE tx ON pt.Ma\_Tuyen = tx.Ma\_Tuyen  INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = cn.Ma\_Chi\_Nhanh  INNER JOIN TAIXE txe ON pt.Ma\_Tai\_Xe = txe.Ma\_Tai\_Xe  INNER JOIN TAIXE\_LIENHE txlh ON txe.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  GROUP BY  pt.Ma\_Phuong\_Tien, pt.Ma\_Tuyen, tx.Ten\_Tuyen,  tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  pt.Ma\_Tai\_Xe, txe.Ho\_Va\_Ten,  pt.Ten\_Phuong\_Tien, pt.Bien\_Kiem\_Soat,  pt.Hang\_San\_Xuat, pt.Nam\_San\_Xuat,  pt.So\_Khung, pt.So\_Cho\_Ngoi; |

Khung 17. Truy vấn VIEW theo dõi chi tiết phương tiện.

|  |
| --- |
| GO  SELECT \* FROM view\_CHITIET\_PHUONGTIEN; |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 10. Kết quả truy vấn VIEW theo dõi chi tiết phương tiện.

Ví dụ 13: Tạo VIEW để theo dõi dữ liệu tài xế và chi nhánh quản lý.

Khung 18. Khởi tạo VIEW theo dõi chi tiết tài xế.

|  |
| --- |
| GO  CREATE VIEW view\_CHITIET\_TAIXE AS  SELECT  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten,  tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  tx.Ngay\_Sinh, tx.Dia\_Chi,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN txlh.Gia\_Tri END, ', ') AS Email,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN txlh.Gia\_Tri END, ', ') AS So\_Dien\_Thoai  FROM TAIXE tx  INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = tx.Ma\_Chi\_Nhanh  INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  GROUP BY  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten,  tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  tx.Ngay\_Sinh, tx.Dia\_Chi; |

Khung 19. Truy vấn VIEW chi tiết tài xế.

|  |
| --- |
| GO  SELECT \* FROM view\_CHITIET\_TAIXE; |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 11. Kết quả truy vấn VIEW chi tiết tài xế.

Ví dụ 14: Tạo VIEW để theo dõi danh sách trạm xe và số tuyến đi qua.

Khung 20. Khởi tạo VIEW chi tiết trạm xe.

|  |
| --- |
| GO  CREATE VIEW view\_CHITIET\_TRAMXE AS  SELECT  tx.Ma\_Tram, tx.Vi\_Tri\_Tram,  STRING\_AGG(lt.Ma\_Tuyen, ', ') WITHIN GROUP (ORDER BY Ma\_Tuyen) AS Tuyen\_Di\_Qua  FROM TRAMXE tx  INNER JOIN LOTRINH lt ON lt.Ma\_Tram = tx.Ma\_Tram  GROUP BY tx.Ma\_Tram, tx.Vi\_Tri\_Tram; |

Khung 21. Truy vấn VIEW chi tiết trạm xe.

|  |
| --- |
| SELECT \* FROM view\_CHITIET\_TRAMXE |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 12. Kết quả truy vấn VIEW chi tiết trạm xe.

Ví dụ 15: Tạo VIEW để theo dõi số tuổi của tài xế.

Khung 22. Khởi tạo VIEW theo dõi tuổi tài xế.

|  |
| --- |
| GO  CREATE VIEW view\_TUOI\_TAIXE AS  SELECT  Ma\_Tai\_Xe, Ho\_Va\_Ten, Ngay\_Sinh,  YEAR(GETDATE()) - YEAR(Ngay\_Sinh) AS Tuoi\_Tai\_Xe  FROM TAIXE |

Khung 23. Truy vấn VIEW theo dõi tuổi tài xế.

|  |
| --- |
| GO  SELECT \* FROM view\_TUOI\_TAIXE |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 13. Kết quả truy vấn VIEW theo dõi tuổi tài xế.

Ví dụ 16: Tạo VIEW để theo dõi số năm còn có thể lao động của tài xế. Biết rằng công ty chỉ cho phép tài xế lao động đến hết năm 55 tuổi.

Khung 24. Khởi tạo VIEW theo dõi số năm còn có thể lao động của tài xế.

|  |
| --- |
| GO  CREATE VIEW view\_TUOI\_NGHIHUU\_TAIXE AS  SELECT  Ma\_Tai\_Xe, Ho\_Va\_Ten, Ngay\_Sinh,  YEAR(GETDATE()) - YEAR(Ngay\_Sinh) AS Tuoi\_Tai\_Xe,  STRING\_AGG(  CAST(DATEDIFF(MONTH, GETDATE(), DATEADD(YEAR, 55, Ngay\_Sinh)) / 12 AS NVARCHAR)  + N' năm ' +  CAST((DATEDIFF(MONTH, GETDATE(), DATEADD(YEAR, 55, Ngay\_Sinh)) % 12) AS NVARCHAR)  + N' tháng ' +  CAST(DATEDIFF(DAY, GETDATE(), DATEADD(YEAR, 55, Ngay\_Sinh)) % 30 AS NVARCHAR)  + N' ngày', ''  ) AS Thoi\_Gian\_Lam\_Viec\_Con\_Lai  FROM TAIXE  GROUP BY Ma\_Tai\_Xe, Ho\_Va\_Ten, Ngay\_Sinh; |

Khung 25. Truy vấn VIEW theo dõi số năm còn có thể lao động của tài xế.

|  |
| --- |
| GO  SELECT \* FROM view\_TUOI\_NGHIHUU\_TAIXE; |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 14. Kết quả thực hiện truy vấn VIEW theo dõi số năm còn có thể lao động của tài xế.

Ví dụ 17: Tạo VIEW để theo dõi số năm còn có thể sử dụng của phương tiện. Biết rằng luật pháp Việt Nam cho phép xe chở khách có niên hạn sử dụng tối đa là 20 năm.

Khung 26. Khởi tạo VIEW theo dõi niên hạn của phương tiện.

|  |
| --- |
| GO  CREATE VIEW view\_NIENHAN\_PHUONGTIEN AS  SELECT  Ma\_Phuong\_Tien, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat,  Nam\_San\_Xuat,  YEAR(GETDATE()) - Nam\_San\_Xuat AS So\_Nam\_Da\_Su\_Dung,  20 - (YEAR(GETDATE()) - Nam\_San\_Xuat) AS So\_Nam\_Con\_Co\_The\_Su\_Dung  FROM PHUONGTIEN; |

Khung 27. Truy vấn VIEW theo dõi niên hạn của phương tiện.

|  |
| --- |
| GO  SELECT \* FROM view\_NIENHAN\_PHUONGTIEN |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 15. Kết quả thực hiện truy vấn VIEW niên hạn của phương tiện.

Ví dụ 18: Tạo VIEW để theo dõi dữ liệu liên hệ của các chi nhánh.

Khung 28. Khởi tạo VIEW theo dõi dữ liệu liên hệ của các chi nhánh.

|  |
| --- |
| GO  CREATE VIEW view\_LIENHE\_CHINHANH AS  SELECT  cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh, cn.Dia\_Chi,  STRING\_AGG(CASE WHEN cnlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN cnlh.Gia\_Tri END, ', ') AS Email,  STRING\_AGG(CASE WHEN cnlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN cnlh.Gia\_Tri END, ', ') AS So\_Dien\_Thoai  FROM CHINHANH cn  INNER JOIN CHINHANH\_LIENHE cnlh ON cn.Ma\_Chi\_Nhanh = cnlh.Ma\_Chi\_Nhanh  GROUP BY cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh, cn.Dia\_Chi; |

Khung 29. Truy vấn VIEW theo dõi dữ liệu liên hệ của các chi nhánh.

|  |
| --- |
| GO  SELECT \* FROM view\_LIENHE\_CHINHANH; |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 16. Kết quả sau khi thực hiện truy vấn VIEW theo dõi liên hệ của các chi nhánh.

Ví dụ 19: Tạo VIEW để theo dõi tổng lượng xe mà mỗi chi nhánh quản lý.

Khung 30. Khởi tạo VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý

|  |
| --- |
| GO  CREATE VIEW view\_PHUONGTIEN\_CHINHANH AS  SELECT  cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  COUNT(DISTINCT pt.Ma\_Phuong\_Tien) AS So\_Phuong\_Tien  FROM PHUONGTIEN pt  INNER JOIN TAIXE tx ON tx.Ma\_Tai\_Xe = pt.Ma\_Tai\_Xe  INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = cn.Ma\_Chi\_Nhanh  GROUP BY cn.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh; |

Khung 31. Truy vấn VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý.

|  |
| --- |
| GO  SELECT \* FROM view\_PHUONGTIEN\_CHINHANH; |

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 17. Kết quả sau khi thực hiện truy vấn VIEW theo dõi tổng lượng xe mà mỗi chi nhánh quản lý.

## **4.2. Tạo INDEX cần thiết cho các bảng**

Khung 32. Khởi tạo chỉ mục (INDEX).

|  |
| --- |
| -- Tạo chỉ mục thông tin của tuyến xe  GO  CREATE INDEX idx\_LOTRINH\_TUYEN  ON LOTRINH(Ma\_Tuyen);  -- Tạo chỉ mục đảm bảo giá trị của biển kiểm soát là duy  -- nhất  GO  CREATE UNIQUE INDEX idx\_BIENKIEMSOAT\_PHUONGTIEN  ON PHUONGTIEN(Bien\_Kiem\_Soat);  -- Tạo chỉ mục đảm bảo 1 phương tiện chỉ được giao cho 1  -- tài xế  GO  CREATE UNIQUE INDEX idx\_TAIXE\_PHUONGTIEN  ON PHUONGTIEN(Ma\_Tai\_Xe);  -- Tạo chỉ mục tìm phương tiện theo tuyến xe  GO  CREATE INDEX idx\_PHUONGTIEN\_TUYEN  ON PHUONGTIEN(Ma\_Tuyen);  -- Tạo chỉ mục tìm tuyến xe theo chi nhánh  GO  CREATE INDEX idx\_TUYENXE\_CHINHANH  ON TUYENXE(Ma\_Chi\_Nhanh);  -- Tạo chỉ mục tìm tuyến xe theo trạm khởi đầu  GO  CREATE INDEX idx\_TRAMKHOIDAU\_TUYENXE  ON TRAMKHOIDAU(Ma\_Tuyen);  -- Tạo chỉ mục tìm tuyến xe theo trạm kết thúc  GO  CREATE INDEX idx\_TRAMKETTHUC\_TUYENXE  ON TRAMKETTHUC(Ma\_Tuyen);  -- Tạo chỉ mục tìm thông tin liên hệ của tài xế  GO  CREATE INDEX idx\_LIENHE\_TAIXE  ON TAIXE\_LIENHE(Ma\_Tai\_Xe);  -- Tạo chỉ mục tìm thông tin của tài xế  GO  CREATE INDEX idx\_THONGTIN\_TAIXE  ON TAIXE(Ma\_Tai\_Xe);  -- Tạo chỉ mục tìm thông tin liên hệ của chi nhánh  GO  CREATE INDEX idx\_LIENHE\_CHINHANH  ON CHINHANH\_LIENHE(Ma\_Chi\_Nhanh);  -- Tạo chỉ mục tìm thông tin của chi nhánh  GO  CREATE INDEX idx\_THONGTIN\_CHINHANH  ON CHINHANH(Ma\_Chi\_Nhanh); |

## **4.3. Tạo STORED PROCEDURE**

### ***4.3.1. STORED PROCEDURE không tham số***

Ví dụ 20: Tạo các STORED PROCEDURE không tham số thực hiện các yêu cầu sau:

a) Lấy danh sách tất cả các tài xế.

b) Lấy danh sách tất cả các phương tiện.

c) Lấy danh sách tất cả các chi nhánh.

d) Lấy danh sách theo dõi tuổi nghỉ hưu của tài xế.

Khung 33. Khởi tạo STORED PROCEDURE không tham số.

|  |
| --- |
| -- a) Lấy danh sách tất cả các tài xế  GO  CREATE PROCEDURE sp\_DANHSACH\_TAIXE  AS  BEGIN  SELECT \* FROM TAIXE;  END;  -- b) Lấy danh sách tất cả các phương tiện  GO  CREATE PROCEDURE sp\_DANHSACH\_PHUONGTIEN  AS  BEGIN  SELECT \* FROM PHUONGTIEN;  END;  -- c) Lấy danh sách tất cả các chi nhánh  GO  CREATE PROCEDURE sp\_DANHSACH\_CHINHANH  AS  BEGIN  SELECT \* FROM CHINHANH;  END;  -- d) Lấy danh sách theo dõi tuổi nghỉ hưu của các tài xế  GO  CREATE PROCEDURE sp\_DANHSACH\_TUOI\_NGHIHUU  AS  BEGIN  SELECT \* FROM view\_TUOI\_NGHIHUU\_TAIXE;  END; |

Khung 34. Cấu trúc cơ bản truy vấn STORED PROCEDURE.

|  |
| --- |
| EXEC tên\_Stored\_Procedure; |

|  |  |
| --- | --- |
|  |  |
|  |  |

Ảnh 18. Kết quả sau khi thực hiện truy vấn STORED PROCEDURE.

### ***4.3.2. STORED PROCEDURE có tham số***

Ví dụ 21: Tạo các STORED PROCEDURE có tham số thực hiện các yêu cầu sau:

a) Lấy danh sách tất cả các tài xế theo chi nhánh.

b) Lấy danh sách tất cả các phương tiện theo hãng sản xuất.

c) Lấy danh sách tất cả phương tiện theo tuyến.

d) Lấy danh sách tất cả tài xế theo tuyến.

e) Lấy danh sách tất cả trạm dừng theo tuyến.

Khung 35. Khởi tạo truy vấn STORED PROCEDURE có tham số.

|  |
| --- |
| -- a) Lấy danh sách tài xế theo chi nhánh  GO  CREATE PROCEDURE sp\_DANHSACH\_TAIXE\_CHINHANH  @Ma\_Chi\_Nhanh NVARCHAR(10)  AS  BEGIN  SELECT \* FROM TAIXE WHERE Ma\_Chi\_Nhanh = @Ma\_Chi\_Nhanh;  END;  -- b) Lấy danh sách phương tiện theo hãng sản xuất  GO  CREATE PROCEDURE sp\_DANHSACH\_PHUONGTIEN\_HANGSX  @Hang\_San\_Xuat NVARCHAR(50)  AS  BEGIN  SELECT  Ma\_Phuong\_Tien, Ten\_Phuong\_Tien,  Bien\_Kiem\_Soat, Hang\_San\_Xuat,  Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM PHUONGTIEN WHERE Hang\_San\_Xuat = @Hang\_San\_Xuat;  END;  -- c) Lấy danh sách phương tiện theo tuyến  GO  CREATE PROCEDURE sp\_DANHSACH\_PHUONGTIEN\_TUYENXE\_MATUYEN  @Ma\_Tuyen NVARCHAR(10)  AS  BEGIN  SELECT  Ma\_Phuong\_Tien, Ten\_Phuong\_Tien,  Bien\_Kiem\_Soat, Hang\_San\_Xuat,  Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM PHUONGTIEN WHERE Ma\_Tuyen = @Ma\_Tuyen;  END;  -- d) Lấy danh sách tài xế theo tuyến  GO  CREATE PROCEDURE sp\_DANHSACH\_TAIXE\_TUYEN\_MATUYEN  @Ma\_Tuyen NVARCHAR(10)  AS  BEGIN  SELECT  tx.Ma\_Tai\_Xe, pt.Ma\_Tuyen,  tx.Ho\_Va\_Ten, tx.Dia\_Chi,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN txlh.Gia\_Tri END, ', ') AS Email,  STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN txlh.Gia\_Tri END, ', ') AS So\_Dien\_Thoai  FROM TAIXE tx  INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  INNER JOIN PHUONGTIEN pt ON tx.Ma\_Tai\_Xe =pt.Ma\_Tai\_Xe  WHERE pt.Ma\_Tuyen = @Ma\_Tuyen  GROUP BY tx.Ma\_Tai\_Xe, pt.Ma\_Tuyen, tx.Ho\_Va\_Ten, tx.Dia\_Chi;  END;  -- e) Lấy danh sách trạm dừng theo tuyến  GO  CREATE PROCEDURE sp\_TUYENXE\_TRAMDUNG  @Ma\_Tuyen NVARCHAR(10)  AS  BEGIN  SELECT  lt.Ma\_Tuyen, lt.Ma\_Tram,  tx.Vi\_Tri\_Tram, lt.Thu\_Tu\_Tram  FROM LOTRINH lt  INNER JOIN TRAMXE tx ON lt.Ma\_Tram = tx.Ma\_Tram  WHERE Ma\_Tuyen = @Ma\_Tuyen;  END; |

Khung 36. Truy vấn STORED PROCEDURE có tham số.

|  |
| --- |
| EXEC sp\_DANHSACH\_TAIXE\_CHINHANH @Ma\_Chi\_Nhanh = 'CN1';  EXEC sp\_DANHSACH\_PHUONGTIEN\_HANGSX @Hang\_San\_Xuat = 'Thaco';  EXEC sp\_DANHSACH\_PHUONGTIEN\_TUYENXE\_MATUYEN @Ma\_Tuyen = 'BUS1';  EXEC sp\_DANHSACH\_TAIXE\_TUYEN\_MATUYEN @Ma\_Tuyen = 'BUS2';  EXEC sp\_TUYENXE\_TRAMDUNG @Ma\_Tuyen = 'BUS11'; |

|  |  |  |  |
| --- | --- | --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. | A screenshot of a computer  AI-generated content may be incorrect. | | A screenshot of a computer  AI-generated content may be incorrect. |
| A screenshot of a computer  AI-generated content may be incorrect. | | A screenshot of a computer  AI-generated content may be incorrect. | |

Ảnh 19. Kết quả sau khi thực hiện STORED PROCEDURE có tham số.

### ***4.3.3. STORED PROCEDURE có đầu ra***

Ví dụ 22: Tạo các STORED PROCEDURE có đầu ra thực hiện các yêu cầu sau:

a) Lấy số lượng trạm dừng của một tuyến xe.

b) Lấy thông tin tài xế.

Khung 37. Khởi tạo STORED PROCEDURE có đầu ra.

|  |
| --- |
| -- a) Lấy số lượng trạm dừng của một tuyến xe  GO  CREATE PROCEDURE sp\_TONG\_TRAMDUNG  @Ma\_Tuyen NVARCHAR(10),  @So\_Luong\_Tram\_Dung INT OUTPUT  AS  BEGIN  SELECT @So\_Luong\_Tram\_Dung = COUNT(\*)  FROM LOTRINH WHERE Ma\_Tuyen = @Ma\_Tuyen;  END;  -- b) Lấy thông tin tài xế  GO  CREATE PROCEDURE sp\_LIENHE\_TAIXE  @Ma\_Tai\_Xe NVARCHAR(10),  @Ho\_Ten NVARCHAR(100) OUTPUT,  @So\_Dien\_Thoai NVARCHAR(15) OUTPUT,  @Email NVARCHAR(100) OUTPUT  AS  BEGIN  SELECT  @Ma\_Tai\_Xe = tx.Ma\_Tai\_Xe,  @Ho\_Ten = tx.Ho\_Va\_Ten,  @So\_Dien\_Thoai = STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại' THEN txlh.Gia\_Tri END, ', '),  @Email = STRING\_AGG(CASE WHEN txlh.Loai\_Lien\_He\_Email\_SDT = 'Email' THEN txlh.Gia\_Tri END, ', ')  FROM TAIXE tx  INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  WHERE @Ma\_Tai\_Xe = tx.Ma\_Tai\_Xe  GROUP BY tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten;  END; |

Khung 38. Truy vấn STORED PROCEDURE có đầu ra.

|  |
| --- |
| GO  DECLARE  @So\_Luong INT,  @Ma\_Tuyen NVARCHAR(10) = 'BUS1'  EXEC sp\_TONG\_TRAMDUNG @Ma\_Tuyen = @Ma\_Tuyen, @So\_Luong\_Tram\_Dung = @So\_Luong OUTPUT;  PRINT N'Số trạm dừng của tuyến có mã "' + @Ma\_Tuyen +'": ' +CAST(@So\_Luong AS NVARCHAR(10));  GO  DECLARE  @Ten NVARCHAR(100),  @So\_Dien\_Thoai NVARCHAR(15),  @Email NVARCHAR(100)  EXEC sp\_LIENHE\_TAIXE @Ma\_Tai\_Xe = N'TX1', @Ho\_Ten = @Ten OUTPUT, @So\_Dien\_Thoai = @So\_Dien\_Thoai OUTPUT,  @Email = @Email OUTPUT;  PRINT N'Tài xế: ' + @Ten + N', Số điện thoại: ' + @So\_Dien\_Thoai + ', Email: ' + @Email; |

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. |  |

Ảnh 20. Kết quả sau khi thực hiện truy vấn STORED PROCEDURE có đầu ra.

## **4.4. Tạo các FUNCTION**

### ***4.4.1. FUNCTION trả về kiểu vô hướng***

Ví dụ 23: Tạo các FUNCTION trả về kiểu vô hướng thực hiện các yêu cầu sau:

a) Tính tổng sổ trạm dừng của 1 tuyến.

b) Tính tổng số tài xế đang làm việc trong hệ thống.

Khung 39. Khởi tạo FUNCTION trả về kiểu vô hướng

|  |
| --- |
| -- a) Tính tổng số trạm dừng của 1 tuyến  GO  CREATE FUNCTION fn\_TRAMDUNG\_TUYENXE (@Ma\_Tuyen NVARCHAR(10))  RETURNS INT  AS  BEGIN  DECLARE @So\_Tram INT;  SELECT @So\_Tram = COUNT(\*) FROM LOTRINH WHERE Ma\_Tuyen = @Ma\_Tuyen;  RETURN @So\_Tram;  END;  -- b) Tính tổng số tài xế đang làm việc trong hệ thống  GO  CREATE FUNCTION fn\_TONG\_TAIXE ()  RETURNS INT  AS  BEGIN  DECLARE @Tong\_Tai\_Xe INT;  SELECT @Tong\_Tai\_Xe = COUNT(\*) FROM TAIXE;  RETURN @Tong\_Tai\_Xe;  END; |

Khung 40. Truy vấn FUNCTION trả về kiểu vô hướng.

|  |
| --- |
| GO  SELECT dbo.fn\_TRAMDUNG\_TUYENXE('BUS1') AS So\_Tram\_Dung;  GO  SELECT dbo.fn\_TONG\_TAIXE() AS Tong\_Tai\_Xe\_Dang\_Lam\_Viec; |

|  |  |
| --- | --- |
|  |  |

Ảnh 21. Kết quả sau khi thực hiện truy vấn FUNCTION trả về kiểu vô hướng.

### ***4.4.2. FUNCTION trả về kiểu bảng***

Ví dụ 24: Tạo các FUNCTION trả về kiểu bảng thực hiện các yêu cầu sau:

a) Lấy dữ liệu số điện thoại của tài xế trong hệ thống.

b) Lấy dữ liệu email của tài xế trong hệ thống.

c) Lấy dữ liệu các tuyến xe.

Khung 41. Khởi tạo FUNCTION trả về kiểu bảng.

|  |
| --- |
| -- a) Lấy dữ liệu số điện thoại của tài xế trong hệ thống  GO  CREATE FUNCTION fn\_SDT\_TAIXE ()  RETURNS TABLE  AS  RETURN  (SELECT  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten,  STRING\_AGG(txlh.Gia\_Tri, ', ') AS So\_Dien\_Thoai  FROM TAIXE tx  INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  WHERE txlh.Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại'  GROUP BY tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten  );  -- b) Lấy dữ liệu email của tài xế trong hệ thống  GO  CREATE FUNCTION fn\_EMAIL\_TAIXE ()  RETURNS TABLE  AS  RETURN  (SELECT  tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten,  STRING\_AGG(txlh.Gia\_Tri, ', ') AS Email  FROM TAIXE tx  INNER JOIN TAIXE\_LIENHE txlh ON tx.Ma\_Tai\_Xe = txlh.Ma\_Tai\_Xe  WHERE txlh.Loai\_Lien\_He\_Email\_SDT = 'Email'  GROUP BY tx.Ma\_Tai\_Xe, tx.Ho\_Va\_Ten  );  -- c) Lấy dữ liệu các tuyến xe  GO  CREATE FUNCTION fn\_THONGTIN\_TUYENXE ()  RETURNS TABLE  AS  RETURN  (SELECT  tx.Ma\_Tuyen, tx.Ten\_Tuyen,  tx.Ma\_Chi\_Nhanh, cn.Ten\_Chi\_Nhanh,  tx.Gio\_Bat\_Dau, tramkd.Vi\_Tri\_Tram AS Tram\_Khoi\_Dau,  tx.Gio\_Ket\_Thuc, tramkt.Vi\_Tri\_Tram AS Tram\_Ket\_Thuc  FROM TUYENXE tx  INNER JOIN CHINHANH cn ON tx.Ma\_Chi\_Nhanh = cn.Ma\_Chi\_Nhanh  INNER JOIN TRAMKHOIDAU kd ON tx.Ma\_Tuyen = kd.Ma\_Tuyen  INNER JOIN TRAMKETTHUC kt ON tx.Ma\_Tuyen = kt.Ma\_Tuyen  INNER JOIN TRAMXE tramkd ON kd.Ma\_Tram = tramkd.Ma\_Tram  INNER JOIN TRAMXE tramkt ON kt.Ma\_Tram = tramkt.Ma\_Tram  ); |

Khung 42. Truy vấn FUNCTION trả về kiểu bảng.

|  |
| --- |
| GO  SELECT \* FROM dbo.fn\_SDT\_TAIXE();  GO  SELECT \* FROM dbo.fn\_EMAIL\_TAIXE();  GO  SELECT \* FROM dbo.fn\_THONGTIN\_TUYENXE(); |

|  |  |
| --- | --- |
|  | |
|  |  |

Ảnh 22. Kết quả sau khi thực hiện truy vấn FUNCTION trả về kiểu bảng.

### ***4.4.3. FUNCTION sử dụng biến bảng***

Ví dụ 25: Tạo các FUNCTION sử dụng biến bảng thực hiện các yêu cầu sau:

a) Sắp xếp số trạm dừng của từng tuyến xe.

b) Sắp xếp số lượng tuyến xe đi qua từng trạm dừng.

Khung 43. Khởi tạo FUNCTION sử dụng biến bảng.

|  |
| --- |
| -- a) Sắp xếp số trạm dừng của từng tuyến xe  GO  CREATE FUNCTION fn\_SOTRAM\_TUYENXE()  RETURNS @Danh\_Sach TABLE (Ma\_Tuyen NVARCHAR(10), Ten\_Tuyen NVARCHAR(100), So\_Tram INT)  AS  BEGIN  INSERT INTO @Danh\_Sach  SELECT tx.Ma\_Tuyen, tx.Ten\_Tuyen, COUNT(DISTINCT lt.Ma\_Tram) AS So\_Tram  FROM TUYENXE tx  INNER JOIN LOTRINH lt ON tx.Ma\_Tuyen = lt.Ma\_Tuyen  GROUP BY tx.Ma\_Tuyen, tx.Ten\_Tuyen;  RETURN;  END;  -- b) Sắp xếp số lượng tuyến xe đi qua của từng trạm  GO  CREATE FUNCTION fn\_TUYENXE\_TRAMDUNG\_SAPXEP()  RETURNS @Danh\_Sach TABLE (Ma\_Tram NVARCHAR(10), So\_Tuyen INT)  AS  BEGIN  INERT INTO @Danh\_Sach  SELECT lt.Ma\_Tram, COUNT(DISTINCT lt.Ma\_Tuyen) AS So\_Tuyen  FROM LOTRINH lt  GROUP BY lt.Ma\_Tram;  RETURN;  END; |

Khung 44. Truy vấn FUNCTION sử dụng biến bảng.

|  |
| --- |
| GO  SELECT \* FROM dbo.fn\_SOTRAM\_TUYENXE() ORDER BY So\_Tram DESC;  GO  SELECT \* FROM dbo.fn\_TUYENXE\_TRAMDUNG\_SAPXEP() ORDER BY So\_Tuyen DESC; |

|  |  |
| --- | --- |
|  |  |

Ảnh 23. Kết quả sau khi thực hiện truy vấn FUNCTION sử dụng biến bảng.

## **4.5. Tạo các TRIGGER để kiểm soát dữ liệu**

### ***4.5.1. TRIGGER theo dõi cập nhật, thay đổi dữ liệu các bảng***

Khung 45. Tạo bảng lưu lịch sử thay đổi dữ liệu.

|  |
| --- |
| GO  CREATE TABLE LICHSUTHAYDOIDULIEU(  ID INT IDENTITY (1, 1) PRIMARY KEY,  Bang\_Tac\_Dong NVARCHAR(50),  Loai\_Tac\_dong NVARCHAR(50),  Ma\_Doi\_Tuong\_Bi\_Tac\_Dong NVARCHAR(10),  Thong\_Tin\_Cu NVARCHAR(MAX),  Thong\_Tin\_Moi NVARCHAR(MAX),  Ngay\_Thay\_Doi DATETIME DEFAULT GETDATE()  ); |

Khung 46. Cài đặt TRIGGER cho các bảng (Ví dụ: Cài đặt cho bảng PHUONGTIEN).

|  |
| --- |
| GO  CREATE TRIGGER trg\_LICHSU\_PHUONGTIEN  ON PHUONGTIEN  AFTER INSERT, UPDATE, DELETE  AS  BEGIN  SET NOCOUNT ON;  -- INSERT  INSERT INTO LICHSUTHAYDOIDULIEU (Bang\_Tac\_Dong, Loai\_Tac\_Dong, Ma\_Doi\_Tuong\_Bi\_Tac\_Dong, Thong\_Tin\_Cu, Thong\_Tin\_Moi, Ngay\_Thay\_Doi)  SELECT  'PHUONGTIEN',  'INSERT',  i.Ma\_Phuong\_Tien,  NULL,  (SELECT Ma\_Phuong\_Tien, Ma\_Tuyen, Ma\_Tai\_Xe, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat, Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM inserted i FOR JSON AUTO),  GETDATE()  FROM inserted i;  -- UPDATE  INSERT INTO LICHSUTHAYDOIDULIEU (Bang\_Tac\_Dong, Loai\_Tac\_Dong, Ma\_Doi\_Tuong\_Bi\_Tac\_Dong, Thong\_Tin\_Cu, Thong\_Tin\_Moi, Ngay\_Thay\_Doi)  SELECT  'PHUONGTIEN',  'UPDATE',  i.Ma\_Phuong\_Tien,  (SELECT Ma\_Phuong\_Tien, Ma\_Tuyen, Ma\_Tai\_Xe, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat, Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM deleted d FOR JSON AUTO),  (SELECT Ma\_Phuong\_Tien, Ma\_Tuyen, Ma\_Tai\_Xe, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat, Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM inserted i FOR JSON AUTO),  GETDATE()  FROM deleted d  JOIN inserted i ON d.Ma\_Phuong\_Tien = i.Ma\_Phuong\_Tien;  -- DELETE  INSERT INTO LICHSUTHAYDOIDULIEU (Bang\_Tac\_Dong, Loai\_Tac\_Dong, Ma\_Doi\_Tuong\_Bi\_Tac\_Dong, Thong\_Tin\_Cu, Thong\_Tin\_Moi, Ngay\_Thay\_Doi)  SELECT  'PHUONGTIEN',  'DELETE',  d.Ma\_Phuong\_Tien,  (SELECT Ma\_Phuong\_Tien, Ma\_Tuyen, Ma\_Tai\_Xe, Ten\_Phuong\_Tien, Bien\_Kiem\_Soat, Hang\_San\_Xuat, Nam\_San\_Xuat, So\_Khung, So\_Cho\_Ngoi  FROM deleted d FOR JSON AUTO),  NULL,  GETDATE()  FROM deleted d;  END; |

Lặp lại các bước cài đặt TRIGGER cho toàn bộ các bảng có trong CSDL. Khi thực hiện cài đặt hoàn tất, toàn bộ lịch sử thay đổi dữ liệu sẽ được lưu vết lại tại bảng LICHSUTHAYDOIDULIEU như ảnh dưới đây.

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 24. Kết quả sau khi cài đặt TRIGGER thành công, toàn bộ lịch sử thay đổi dữ liệu sẽ được lưu vết tại bảng LICHSUTHAYDOIDULIEU.

### ***4.5.2. Các TRIGGER khác***

Khung 47. Khởi tạo TRIGGER ngăn xóa tài xế khi còn dữ liệu phương tiện liên quan.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_PHUONGTIEN\_TAIXE  ON TAIXE  INSTEAD OF DELETE  AS  BEGIN  IF EXISTS (SELECT 1 FROM deleted d INNER JOIN PHUONGTIEN pt ON d.Ma\_Tai\_Xe = pt.Ma\_Tai\_Xe)  BEGIN  RAISERROR (N'Tài xế đang có phương tiện, không thể xóa', 16, 1);  RETURN;  END  DELETE FROM TAIXE WHERE Ma\_Tai\_Xe IN (SELECT Ma\_Tai\_Xe from deleted);  END; |

Khung 48. Khởi tạo TRIGGER kiểm tra biển kiểm soát của phương tiện.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_BIENKIEMSOAT  ON PHUONGTIEN  AFTER INSERT  AS  BEGIN  IF EXISTS (SELECT 1 FROM inserted WHERE Bien\_Kiem\_Soat NOT LIKE '[0-9][0-9][A-Z]-[0-9][0-9][0-9].[0-9][0-9]')  BEGIN  RAISERROR (N'Biển số xe không hợp lệ!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

Khung 49. Khởi tạo TRIGGER ngăn chặn thay đổi thông tin mã tuyến.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_MATUYEN  ON TUYENXE  INSTEAD OF UPDATE  AS  BEGIN  IF UPDATE(Ma\_Tuyen)  BEGIN  RAISERROR (N'Không được phép cập nhật mã tuyến!', 16, 1);  RETURN;  END  UPDATE TUYENXE  SET Ten\_Tuyen = i.Ten\_Tuyen, Gio\_Bat\_Dau = i.Gio\_Bat\_Dau, Gio\_Ket\_Thuc = i.Gio\_Ket\_Thuc  FROM TUYENXE t INNER JOIN inserted i ON t.Ma\_Tuyen = i.Ma\_Tuyen;  END; |

Khung 50. Khởi tạo TRIGGER chặn trùng lặp số điện thoại cho tài xế và chi nhánh.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_SDT\_TAIXE  ON TAIXE\_LIENHE  AFTER INSERT  AS  BEGIN  IF EXISTS (SELECT Gia\_Tri FROM inserted i WHERE Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại'  GROUP BY Gia\_Tri HAVING COUNT(\*) > 1)  BEGIN  RAISERROR (N'Số điện thoại đã tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  END  END;  GO  CREATE TRIGGER trg\_CHECK\_SDT\_CHINHANH  ON CHINHANH\_LIENHE  AFTER INSERT  AS  BEGIN  IF EXISTS (SELECT Gia\_Tri FROM inserted i WHERE Loai\_Lien\_He\_Email\_SDT = N'Số điện thoại'  GROUP BY Gia\_Tri HAVING COUNT(\*) > 1)  BEGIN  RAISERROR (N'Số điện thoại đã tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

Khung 51. Khởi tạo TRIGGER kiểm tra số chỗ ngồi của phương tiện.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_SOCHONGOI  ON PHUONGTIEN  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT 1 FROM inserted WHERE So\_Cho\_Ngoi < 16)  BEGIN  RAISERROR (N'Số chỗ ngồi không hợp lệ, phải lớn hơn 16!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

Khung 52. Khởi tạo TRIGGER chặn trùng lặp vị trí của trạm xe.

|  |
| --- |
| GO  CREATE TRIGGER trg\_CHECK\_VITRI\_TRAMDUNG  ON TRAMXE  AFTER INSERT  AS  BEGIN  IF EXISTS (SELECT 1 FROM inserted i INNER JOIN TRAMXE t ON i.Vi\_Tri\_Tram = t.Vi\_Tri\_Tram)  BEGIN  RAISERROR (N'Vị trí trạm đã tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

**CHƯƠNG 5:**

# **BẢO MẬT VÀ QUẢN TRỊ**

## **5.1. Tạo người dùng**

Khung 53. Khởi tạo người dùng trong CSDL.

|  |
| --- |
| GO  USE master;  -- Tạo tài khoản cho Admin  CREATE LOGIN USER\_ADMIN WITH PASSWORD = 'Admin123@';  -- Tạo tài khoản cho nhân viên level 1  CREATE LOGIN USER\_EMP\_LV1 WITH PASSWORD = 'EMP123@1';  -- Tạo tài khoản cho nhân viên level 2  CREATE LOGIN USER\_EMP\_LV2 WITH PASSWORD = 'EMP123@2';  -- Tạo tài khoản cho nhân viên level 3  CREATE LOGIN USER\_EMP\_LV3 WITH PASSWORD = 'EMP123@3';  -- Tạo user trong DB  USE Transerco\_Bus\_Management\_DB;  CREATE USER USER\_ADMIN FOR LOGIN USER\_ADMIN;  CREATE USER USER\_EMP\_LV1 FOR LOGIN USER\_EMP\_LV1;  CREATE USER USER\_EMP\_LV2 FOR LOGIN USER\_EMP\_LV2;  CREATE USER USER\_EMP\_LV3 FOR LOGIN USER\_EMP\_LV3; |

## **5.2. Phân cấp quyền truy cập của người dùng**

Quyền truy cập của từng đối tượng người dùng được phân cấp như sau:

- Đối với tài khoản Admin: Có toàn quyền đọc, thay đổi dữ liệu hoặc cấu trúc bảng trên CSDL. Đồng thời có quyền phân quyền cho các đối tượng người dùng khác.

- Đối với tài khoản nhân viên mức 1: Có thể đọc dữ liệu, thay đổi dữ liệu hoặc cấu trúc bảng trên CSDL. Không có quyền phân quyền cho các đối tượng người dùng khác.

- Đối với tài khoản nhân viên mức 2: Có thể đọc và thay đổi dữ liệu trên CSDL, nhưng không thể thay đổi cấu trúc bảng.

- Đối với tài khoản nhân viên mức 3: Chỉ có thể đọc dữ liệu trên CSDL mà không thể thay đổi chúng.

Để thực hiện cài đặt phân quyền như trên, ta dùng các câu lệnh sau:

Khung 54. Cài đặt quyền cho từng đối tượng người dùng.

|  |
| --- |
| -- Tài khoản Admin - Full Access - db\_owner  ALTER ROLE db\_owner ADD MEMBER USER\_ADMIN;  -- Tài khoản level 1 - db\_ddladmin, db\_datawriter,  -- db\_datareader  ALTER ROLE db\_ddladmin ADD MEMBER USER\_EMP\_LV1;  ALTER ROLE db\_datawriter ADD MEMBER USER\_EMP\_LV1;  ALTER ROLE db\_datareader ADD MEMBER USER\_EMP\_LV1;  -- Tài khoản level 2 - db\_datawriter, db\_datareader  ALTER ROLE db\_datawriter ADD MEMBER USER\_EMP\_LV2;  ALTER ROLE db\_datareader ADD MEMBER USER\_EMP\_LV2;  -- Tài khoản level 3 - db\_datareader  ALTER ROLE db\_datareader ADD MEMBER USER\_EMP\_LV3; |

## **5.3. Quản lý sao lưu và phục hồi dữ liệu**

***5.3.1. Sao lưu CSDL trên thiết bị chạy hệ điều hành Windows***

Trên các thiết bị chạy hệ điều hành Windows, để tự động sao lưu dữ liệu, có thể sử dụng SQL Server Agent (yêu cầu SQL Server Enterprise Edition). Quá trình thực hiện bao gồm các bước sau:

Bảng 2. Quy trình cài đặt backup tự động cho CSDL.

|  |  |  |
| --- | --- | --- |
| Bước | Hình ảnh minh hoạ | Nội dung |
| 1 |  | Tại thẻ **Object Explorer** => Chọn **Server Objects** =>Chọn **Backup Device** => Chọn **New Backup Device** |
| 2 |  | Tại cửa sổ **Backup Device** =>Cài đặt tên tại **Device name** =>Lựa chọn vị trí lưu bản backup =>Tại cửa sổ **Locate Database Files** => Đặt tên cho file tại **File name** |
| 3 |  | Sau khi tạo **Backup Device** => Tại CSDL cần backup, nhấn chuột phải => Chọn **Tasks** => Chọn **Backup** |
| 4 |  | Tại cửa sổ **Back Up Database** => Xoá đường dẫn mặc định => Chọn **Add** => Tại cửa sổ **Select Backup Destination** => Chọn **Backup device** => Chọn **OK** => Chọn **OK** |
| 5 |  | Sau khi chọn **OK** => Nếu backup thành công sẽ hiện thông báo như ảnh minh hoạ => Chọn **OK** để kết thúc. |
| 6 |  | Tại thẻ **Object Explorer** => Chọn **Management** => Chọn **Maintenance Plans** => Chọn **Maintenance Plan Wizard** |
| 7 |  | Tại cửa sổ **Maintenance Plan Wizard** => Đặt tên cho plan tại **Name** => Chọn **Separate schedules for each task** => Tại mục **Schedule** => Chọn **Change** để cài đặt lịch tự động backup |
| 8 |  | Tại cửa sổ **New Job Schedule** => Tại mục **Frequency**, có thể cài đặt tần suất backup dữ liệu => Tại mục **Daily frequency**, có thể cài đặt giờ thực hiện backup trong ngày => Tại mục **Duration**, có thể cài đặt ngày ngừng backup => Lịch backup tự động sau khi cài đặt được mô tả tại mục **Summary** |
| 9 |  | Sau khi cài đặt lịch backup tự động thành công => Tại cửa sổ **Maintenance Plan Wizard** => Chọn **Next** => Khi được hỏi **Select Maintenance Tasks** => Chọn **Back Up Database (Full)** => Chọn **Next** |
| 10 |  | Khi xuất hiện **Define Back Up Database (Full) Task** => Tại thẻ **General**, mục **Database(s)** => Chọn đúng database cần backup tại mục **These databases** => Chọn **OK** |
| 11 |  | Tại thẻ **Destination** => Chọn **Back up databases across one or more files** => Khi hiện cửa sổ **Select Backup Destination** => Chọn **Backup device** |
| 12 |  | Tại thẻ **Options** => Chọn **Verify backup integrity** => Chọn **Next** |
| 13 |  | Sau khi chạy thành công => Hiện thông báo như ảnh minh hoạ => Chọn **Close** |
| 14 |  | Để cài đặt tính năng gửi thông báo về email, tại **SQL Server Agent** => Chọn **Properties** => Chọn **Alert System** => Chọn **Enable mail profile** => Chọn profile tại **Mail profile** => Chọn **OK** |
| 15 |  | Tại **Object Explorer** => Chọn **SQL Server Agent** => Chọn **Operators** => Chọn **New Operator** => Nhập tên tại **Name** => Chọn **Enabled** => Nhập địa chỉ email nhận thông báo tại **E-mail name** => Chọn **OK** |
| 16 |  | Tại **Object Explorer** => Chọn **SQL Server Agent** => Chọn **Jobs** => Chọn **Subplan** vừa tạo => Chọn **Properties** => Chọn **Email** => Cài đặt khi nào thì nhận email thông báo => Chọn **Write to the Windows Application event log** => Chọn **OK** |
| 17 |  | Tại **Object Explorer** => Chọn **SQL Server Agent** => Chọn **Jobs** => Chọn **Subplan** vừa tạo => Chọn **Start Job at Step…** => Nếu job khởi động thành công sẽ hiện thông báo như ảnh |

***5.3.2. Sao lưu CSDL trên thiết bị chạy hệ điều hành macOS***

Do SQL Server Agent hiện chỉ hỗ trợ hệ điều hành Windows mà không hỗ trợ trên macOS hoặc Linux. Vì vậy, trên các thiết bị chạy hệ điều hành macOS, không thể sử dụng SQL Server Agent để lên lịch backup tự động như trên Windows. Thay vào đó, người dùng phải thực hiện backup thủ công theo các bước sau:

Bảng 3. Quy trình backup thủ công cho thiết bị chạy hệ điều hành macOS.

|  |  |
| --- | --- |
| **Bước** | **Cách thực hiện** |
| 1 | Chạy câu lệnh dưới đây trong Azure Data Studio:  BACKUP DATABASE Transerco\_Bus\_Management\_DB  TO DISK = '/var/opt/mssql/backup/Transerco\_Backup.bak'  WITH FORMAT, COMPRESSION, INIT; |
| 2 | Sau khi chạy câu lệnh trên, mở **Terminal** và chạy lệnh sau để liệt kê các file backup trong container:  docker exec -it sql2022 ls -lh /var/opt/mssql/backup  Nếu thấy file backup vừa tạo ở bước 1 nghĩa là CSDL đã được backup. |
| 3 | Sao chép file backup từ Docker ra máy macOS bằng cách gõ lệnh sau trong **Terminal**:  docker cp sql2022:/var/opt/mssql/backup/Transerco\_Backup.bak ~/Documents/Transerco\_Backup.bak |

# **KẾT LUẬN**

**Quá trình thực hiện:**

Quá trình thực hiện bài tập lớn đề tài “Thiết kế và triển khai cơ sở dữ liệu cho hệ thống quản lý xe buýt liên tỉnh” với sự hướng dẫn của Thạc sĩ Nguyễn Ngọc Ân, đã giúp nhóm chúng tôi có cơ hội tiếp cận một cách thực tế và khoa học với các vấn đề trong quản lý dữ liệu. Để hoàn thành đề tài, nhóm chúng tôi đã trải qua các giai đoạn chính sau:

- Tìm hiểu và phân tích bài toán: Nhóm đã nghiên cứu kĩ càng về mô hình tổ chức của một đơn vị cung cấp dịch vụ vận tải công cộng từ thực tế (Transerco), qua đó xác định các thực thể quan trọng trong bài toán quản lý như tuyến xe, trạm dừng, phương tiện, tài xế, chi nhánh và thành phố. Từ đó xây dựng một hệ thống có thể đáp ứng tối đa các yêu cầu khi đưa vào vận hành thực tế.

- Thiết kế các mô hình ERD và mô hình quan hệ: Từ những dữ liệu thu thập được qua quá trình tìm hiểu thực tế, nhóm đã xây dựng mô hình ERD cho bài toán, nhằm trực quan hoá cách mà các thực thể liên kết với nhau. Việc xác định các mối quan hệ giữa các thực thể là bước quan trọng để đảm bảo tính logic và hiệu quả của hệ thống. Từ mô hình ERD, nhóm đã áp dụng các quy tắc chuyển đổi để đảm bảo các bảng trong CSDL được tổ chức một cách khoa học, không có dữ liệu dư thừa, đồng thời duy trì tính toàn vẹn của dữ liệu.

- Chuẩn hoá và cài đặt CSDL bằng SQL: Nhóm đã thực hiện việc chuẩn hoá các bảng dữ liệu đến chuẩn 3NF để loại bỏ dư thừa và tránh xung đột dữ liệu. Sau đó, nhóm tiếp tục sử dụng các câu lệnh SQL để tạo CSDL.

- Tìm hiểu và triển khai các câu lệnh nâng cao: Sau khi xây dựng CSDL bằng SQL, nhóm đã thực hiện việc triển khai các câu lệnh nâng cao như JOIN, GROUP BY, HAVING, SUBQUERY, tạo View, tạo Stored Procedure, tạo Trigger,… Quá trình này giúp nhóm hiểu sâu hơn về cách tổ chức và quản lý dữ liệu.

**Kết quả đạt được:**

Sản phẩm thu được sau quá trình thực hiện đề tài có thể truy cập thông qua hai đường dẫn sau:

- Github: <https://github.com/MinhNghia2802/BTL-HQT-CSDL>

- Onedrive: <https://dainamedu-my.sharepoint.com/:f:/g/personal/1771020505_dnu_edu_vn/EtHRBVsKb99DhSUVhd_J4XUBPyYZDsGlFXRGbdE5QzWDdw?e=Fa7JM1>

Về ưu điểm của hệ thống đã được triển khai:

- Hệ thống được thiết kế chặt chẽ theo mô hình chuẩn hoá 3NF, đảm bảo tính toàn vẹn dữ liệu và tránh trùng lặp.

- Các ràng buộc khoá chính và khoá ngoại được thiết lập rõ ràng, giúp duy trì tính nhất quán của dữ liệu.

- Cung cấp nhiều truy vấn giúp tối ưu hoá việc tìm kiếm và phân tích dữ liệu.

- Sử dụng các kỹ thuật nâng cao như Stored Procedure và Trigger giúp kiểm soát dữ liệu một cách hiệu quả và tăng tính tự động hoá.

- Hệ thống được thiết lập tự động backup giúp ngăn ngừa rủi ro mất mát dữ liệu.

Về nhược điểm của hệ thống đã được triển khai:

- Do quy mô của một bài tập lớn và số lượng nhân sự của nhóm có hạn, nên hệ thống chưa được kiểm thử trên dữ liệu thực tế với quy mô lớn.

- Một số truy vấn có thể cần tối ưu hoá thêm để đảm bảo tốc độ phản hồi tốt nhất.

- Hệ thống chưa được thiết kế giao diện UI/UX, hiện tại việc tương tác với hệ thống chủ yếu dựa vào truy vấn SQL.

**Công việc trong tương lai:**

Trong tương lai, nếu có đủ thời gian và các nguồn lực cần thiết, nhóm dự định:

- Thu thập thêm dữ liệu từ thực tế, xây dựng bộ dữ liệu mẫu quy mô lớn hơn để kiểm thử hiệu suất của hệ thống.

- Thiết kế giao diện người dùng bằng các HTML, CSS, JavaScript hoặc sử dụng các nền tảng như Streamlit để giúp người dùng dễ dàng thao tác với hệ thống.

Tổng kết lại, đề tài đã giúp nhóm có cơ hội áp dụng những kiến thức đã học trong học phần Hệ quản trị cơ sở dữ liệu vào thực tế. Dù còn một số hạn chế, nhưng sản phẩm cuối cùng đã đạt được những yêu cầu đề ra và có tiềm năng phát triển hơn nữa trong tương lai.

# **PHÂN CÔNG NHIỆM VỤ**

Bảng 4. Bảng phân công nhiệm vụ.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | *Tham gia: X* | | *Không tham gia: O* | | |
| **STT** | **Đầu việc** | | | **Thành viên** | | | |
| **Nghĩa** | | **Lương** | **Lan** |
| 1 | Xây dựng bối cảnh giả định, mô tả bài toán và yêu cầu cần đạt được | | | o | | o | x |
| 2 | Xác định thực thể và các thuộc tính tương ứng | | | o | | x | o |
| 3 | Xác định liên kết và kiểu liên kết giữa các thực thể | | | o | | x | x |
| 4 | Xây dựng mô hình ERD | | | x | | o | o |
| 5 | Chuyển hoá mô hình ERD sang mô hình quan hệ | | | x | | o | o |
| 6 | Xác định các ràng buộc | | | x | | o | o |
| 7 | Xác định các phụ thuộc hàm | | | x | | o | o |
| 8 | Chuẩn hoá dữ liệu theo chuẩn 3NF | | | x | | o | o |
| 9 | Tạo cơ sở dữ liệu và các bảng | | | o | | x | x |
| 10 | Cài đặt các ràng buộc | | | o | | x | x |
| 11 | Cài đặt PK và FK | | | o | | x | x |
| 12 | Nhập dữ liệu cho các bảng | | | x | | x | x |
| 13 | Truy vấn cơ bản (SELECT, INSERT, UPDATE, DELETE) | | | o | | x | x |
| 14 | Truy vấn nâng cao (INNER JOIN, GROUP BY, HAVING, SUBQUERY) | | | x | | o | o |
| 15 | Tạo view cho CSDL | | | x | | o | o |
| 16 | Tạo index cần thiết cho các bảng | | | x | | o | o |
| 17 | Tạo Stored Procedure | | | x | | o | o |
| 18 | Tạo các Function | | | x | | o | o |
| 19 | Tạo các Trigger | | | x | | o | o |
| 20 | Tạo người dùng và cấp quyền truy cập của người dùng | | | o | | x | x |
| 21 | Cài đặt sao lưu tự động cho CSDL | | | x | | x | o |
| 22 | Thiết kế slide thuyết trình báo cáo | | | o | | x | x |
| TỶ LỆ THAM GIA VÀO ĐỀ TÀI | | | | 59,09% | | 45,45% | 45,45% |

# **ĐÁNH GIÁ NỘI BỘ**

Bảng 5. Bảng đánh giá thái độ và tinh thần tham gia.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thành viên** | **MSV** | **Đánh giá thái độ và tinh thần tham gia (…/10)** | | | |
| **Nghĩa** | **Lương** | **Lan** | **TB** |
| 1 | Nghĩa | 1771020505 | x | 10 | 9 | **9.5** |
| 2 | Lương | 1771020444 | 10 | x | 8 | **9** |
| 3 | Lan | 1771020413 | 9 | 9 | x | **9** |

Bảng 6. Bảng đánh giá chất lượng công việc.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thành viên** | **MSV** | **Đánh giá chất lượng công việc (…/10)** | | | |
| **Nghĩa** | **Lương** | **Lan** | **TB** |
| 1 | Nghĩa | 1771020505 | x | 10 | 10 | **10** |
| 2 | Lương | 1771020444 | 9 | x | 9 | **9** |
| 3 | Lan | 1771020413 | 10 | 9 | x | **9.5** |

# **TÀI LIỆU THAM KHẢO**

[1]. ThS. Nguyễn Ngọc Ân (2025), *Tài liệu học tập học phần Hệ quản trị cơ sở dữ liệu*, Khoa Công nghệ - Thông tin, Trường Đại học Đại Nam.

[2]. ThS. Nguyễn Ngọc Ân (2025), *Bài giảng học phần Hệ quản trị cơ sở dữ liệu*, Khoa Công nghệ - Thông tin, Trường Đại học Đại Nam.