

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN



BÀI TẬP LỚN 1 – THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

MÔN HỌC: CƠ SỞ DỮ LIỆU PHÂN TÁN

ĐỀ TÀI: Cửa hàng cầu lông đa chi nhánh

Giảng viên giảng dạy: Nguyễn Minh Nhựt

Lớp: IS211.P21

Nhóm sinh viên thực hiện

- | | |
|--------------------------|----------|
| 1. Nguyễn Thị Hải Hà | 22520372 |
| 2. Nguyễn Thị Thanh Hiếu | 22520443 |
| 3. Nguyễn Hồng Nhật Long | 22520816 |
| 4. Phạm Đức Mạnh | 22520852 |

TPHCM, ngày 03 tháng 06 năm 2025

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin được bày tỏ lòng biết ơn sâu sắc đến toàn thể quý thầy cô Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM, đặc biệt là quý thầy cô Khoa Hệ thống Thông tin, những người đã luôn tận tâm trong giảng dạy và tạo điều kiện thuận lợi cho sinh viên được tiếp cận với kiến thức tiên tiến, môi trường học tập hiện đại và các công nghệ mới trong lĩnh vực khoa học dữ liệu và công nghệ thông tin. Trong suốt quá trình học tập và nghiên cứu, chúng em không chỉ được trang bị kiến thức nền tảng vững chắc mà còn được truyền cảm hứng học thuật và tinh thần đổi mới sáng tạo.

Đặc biệt, nhóm xin gửi lời cảm ơn chân thành và lòng kính trọng sâu sắc đến **Thầy Nguyễn Minh Nhựt**, người đã trực tiếp hướng dẫn và đồng hành cùng chúng em trong suốt quá trình thực hiện báo cáo. Thầy không chỉ hỗ trợ về mặt chuyên môn với sự tận tụy, nghiêm túc và trách nhiệm, mà còn luôn khuyến khích chúng em suy nghĩ độc lập, khai thác nhiều nguồn tư liệu. Những đóng góp quý báu từ thầy đã giúp chúng em mở rộng tư duy, rèn luyện khả năng nghiên cứu và tiếp cận vấn đề một cách khoa học, logic và sáng tạo hơn.

Chúng em cũng xin chân thành cảm ơn toàn thể các thành viên trong lớp đã cùng nhau học tập, chia sẻ tài liệu, kinh nghiệm và động viên nhau trong suốt quá trình qua. Sự gắn kết và tinh thần tập thể chính là yếu tố then chốt giúp môi trường học tập trở nên tích cực, hỗ trợ lẫn nhau cùng tiến bộ.

Mặc dù đã nỗ lực hết mình, nhưng với kinh nghiệm còn hạn chế, nên báo cáo không tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được những nhận xét, phản hồi và góp ý chân thành từ quý thầy để không ngừng nâng cao năng lực bản thân, rút kinh nghiệm cho những dự án học thuật và nghề nghiệp trong tương lai.

Chúng em xin chân thành cảm ơn!

TPHCM, ngày 03 tháng 06 năm 2025

Nguyễn Thị Hải Hà – 22520372

Nguyễn Thị Thanh Hiếu – 22520443

Nguyễn Hồng Nhật Long - 22520816

Phạm Đức Mạnh - 22520852

MỤC LỤC

1. THIẾT KẾ HỆ THỐNG PHÂN TÁN BÁN HÀNG ĐA CHI NHÁNH	7
1.1. Mô tả bài toán	7
1.2. Lược đồ dữ liệu tổng thể	7
1.3. Phân mảnh dữ liệu	8
1.3.1. Phân mảnh ngang	8
1.3.2. Phân mảnh dọc	8
1.3.3. Nhân bản (Replicated)	9
1.4. Phân bố dữ liệu theo máy ảo	9
1.4.1. Quyền truy cập đề xuất	9
1.4.2. Phân quyền: (tự thêm số vô TRUONGCN với NHANVIEN nha)	12
1.4.3. Tạo Database link	13
1.5. Câu truy vấn	14
1.5.1. (Thực hiện ở CN1) Giám đốc tìm nhân viên xuất sắc nhất năm 2024 của mỗi chi nhánh (doanh thu cao nhất)	14
1.5.2. (Thực hiện ở CN1) Giám đốc thống kê doanh thu theo từng chi nhánh trong năm 2024, sau đó tính tổng doanh thu của hệ thống.	15
1.5.3. (Thực hiện ở CN1) Giám đốc tìm ra khách hàng tiềm năng đã mua hàng từ 2 chi nhánh trở lên	16
1.5.4. (Thực hiện ở CN1) Giám đốc thống kê mức lương trung bình trả cho một nhân viên tại mỗi chi nhánh	16
1.5.5. (Thực hiện ở CN2) Nhân viên tìm các sản phẩm còn hàng thuộc danh mục 'Vợt cầu lông' có giá trên 2 triệu dưới 3 triệu.	17
1.5.6. (Thực hiện ở CN2) Giám đốc tìm khách hàng đã mua tất cả sản phẩm từ hãng VNB	19
1.5.7. (Thực hiện ở CN2) Giám đốc tìm ra chi nhánh có mức tăng trưởng doanh thu tốt nhất năm 2024 (doanh thu cả năm so với doanh thu 6 tháng đầu tiên)	20
1.5.8. (Thực hiện ở CN3) Quản lý liệt kê số lượng các nhân viên làm việc tại từng chi nhánh	21

1.5.9. (Thực hiện ở CN3) Nhân viên tìm sản phẩm hết hàng ở CN3 nhưng còn hàng ở CN1 và CN2.	21
1.5.10. (Thực hiện tại CN3) Giám đốc thống kê Top 5 sản phẩm bán chạy nhất toàn hệ thống	22
2. VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆN	23
2.1. Function	23
2.1.1. Function thống kê mua hàng khách hàng trên toàn hệ thống	23
2.2. Procedure	25
2.2.1. Procedure Thêm 1 khách hàng mới bất kì vào 1 chi nhánh bất kì sẽ đồng bộ với các chi nhánh còn lại	25
2.3. Ràng buộc toàn vẹn	26
2.3.1. Bảng tầm ảnh hưởng.....	26
3. CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHÂN TÁN.....	28
3.1. Lost update	28
3.1.1. Mô tả tình huống	28
3.1.2. Minh họa.....	29
3.1.3. Giải pháp.....	29
3.2. Phantom Read.....	30
3.2.1. Mô tả tình huống	30
3.2.2. Minh họa.....	31
3.2.3. Giải pháp.....	31
3.3. Dirty Read.....	32
3.4. Unrepeatable Read.....	32
3.4.1. Mô tả tình huống	32
3.4.2. Minh họa.....	33
3.4.3. Giải pháp.....	33
3.5. Deadlock.....	34
3.5.1. Mô tả tình huống	34

3.5.2. Minh họa.....	34
3.5.3. Giải pháp.....	35
4. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN	35
4.1. Câu truy vấn SQL chưa tối ưu	35
4.2. Thực hiện EXPLAIN câu truy vấn	37
4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán.....	38
4.3.1. Tối ưu hoá câu truy vấn cục bộ	38
4.3.2. Câu truy vấn rút gọn trên các mảnh.....	45
4.4. Viết lại câu truy vấn rút gọn trên các mảnh.....	45
5. CHỈ MỤC TRONG ORACLE	46
5.1. Lý thuyết về các chỉ mục.....	47
5.1.1. Chỉ mục B-tree (B-tree Index).....	47
5.1.2. Chỉ mục Bitmap (Bitmap Index)	49
5.1.3. Index-Organized Table (IOT)	49
5.1.4. Chỉ mục đảo ngược khoá (Reverse Key Index)	50
5.1.5. Chỉ mục dựa trên hàm (Function-Based Index)	51
5.1.6. Partitioned Index (Local & Global).....	52
5.1.7. Invisible Index	53
5.1.8. Chỉ mục tổng hợp (Composite/Concatenated Index)	54
5.1.9. Auto Indexing (Automatic Indexes)	54
5.1.10. Sharded Index	55
5.1.11. Chỉ mục miền ứng dụng (Domain/Application Index).....	56
5.1.12. Hash Clustered Index	56
5.2. Triển khai các loại chỉ mục trong truy vấn phân tán	57
5.2.1. Sử dụng chỉ mục để tối ưu câu truy vấn	58
6. PHÂN CÔNG CÔNG VIỆC	61
7. TÀI LIỆU THAM KHẢO	62

DANH MỤC BẢNG

Bảng 1. Phân bố dữ liệu.....	9
Bảng 2. Tạo bảng.....	9
Bảng 3. Phân quyền.....	12
Bảng 4. Tạo Database link.....	13
Bảng 5. Minh họa tình huống Lost update	29
Bảng 6. Giải pháp cho tình huống Lost update	30
Bảng 7. Minh họa cho tình huống Phantom Read.....	31
Bảng 8. Giải pháp cho tình huống Phantom Read.....	32
Bảng 9. Minh họa tình huống Unrepeatable Read	33
Bảng 10. Giải pháp cho tình huống Unrepeatable Read	33
Bảng 11. Bảng viết tắt	38
Bảng 12. Phân công công việc	61

DANH MỤC HÌNH ẢNH

Hình 1-1. Kết quả câu truy vấn số 1	15
Hình 1-2. Kết quả câu truy vấn số 2	15
Hình 1-3. Kết quả câu truy vấn số 3	16
Hình 1-4. Kết quả câu truy vấn số 4	17
Hình 1-5. Kết quả câu truy vấn số 5	19
Hình 1-6. Kết quả câu truy vấn số 7	21
Hình 1-7. Kết quả câu truy vấn số 8	21
Hình 1-8. Kết quả câu truy vấn số 9	22
Hình 1-9. Kết quả câu truy vấn số 10	23
Hình 4-1. Đồ thị truy vấn	38
Hình 4-2. Cây truy vấn ban đầu	39
Hình 4-3. Cây truy vấn tối ưu toàn cục	40
Hình 4-4. Thời gian truy vấn khi chưa tối ưu	43
Hình 4-5. Thời gian truy vấn sau khi tối ưu	44
Hình 4-6. Cây truy vấn rút gọn trên các mảnh	45
Hình 4-7. Kết quả Explain câu truy vấn.	46
Hình 5-1. Kết quả sau khi explain câu truy vấn.	59
Hình 5-2. Kết quả EXPLAIN câu truy vấn sau khi sử dụng chỉ mục.	60

1. THIẾT KẾ HỆ THỐNG PHÂN TÁN BÁN HÀNG ĐA CHI NHÁNH

1.1. Mô tả bài toán

- Hệ thống quản lý các chi nhánh của chuỗi cửa hàng **Bán hàng đa chi nhánh** với 3 chi nhánh:
 - + **CN1:** TP.HCM
 - + **CN2:** Hà Nội
 - + **CN3:** Đà Nẵng

1.2. Lược đồ dữ liệu tổng thể

CHINHANH(MACHINHANH, TENCHINHANH, DIACHI, SODIENTHOAI)

Tân từ: Mỗi chi nhánh có mã chi nhánh (MACHINHANH) là duy nhất để phân biệt các chi nhánh, tên chi nhánh (TENCHINHANH), địa chỉ (DIACHI), và số điện thoại (SODIENTHOAI). Bảng này thường đặt tại site trung tâm để quản trị toàn bộ hệ thống.

KHACHHANG(MAKH, HOTEN, DIACHI, SODIENTHOAI, TONGTIEN)

Tân từ: Mỗi khách hàng có mã khách hàng (MAKH) duy nhất. Tên khách hàng (HOTEN), địa chỉ (DIACHI), số điện thoại (SODIENTHOAI), tổng tiền đã chi tiêu (TONGTIEN). Bảng này được phân mảnh ngang theo vùng miền.

SANPHAM(MASP, TENSANPHAM, DANHMUC, THUONGHIEU, GIANHAP, GIABAN, ISDELETE)

Tân từ: Mỗi sản phẩm có mã sản phẩm (MASP) duy nhất. Bảng này được nhân bản tại tất cả các chi nhánh để truy vấn nhanh, không phụ thuộc từ xa. ISDELETE đánh dấu sản phẩm ngừng kinh doanh.

NHANVIEN(MANV, HOTEN, GIOITINH, NGAYSINH, DIACHI, SODIENTHOAI, NGAYVAOLAM, CHUCVU, LUONG, MACHINHANH, ISDELETE)

Tân từ: Nhân viên có mã (MANV) duy nhất. Bảng này được phân mảnh dọc không dư thừa:

- **NHANVIEN_PUBLIC(MANV, HOTEN, GIOITINH, NGAYSINH, DIACHI, SODIENTHOAI, NGAYVAOLAM, CHUCVU, MACHINHANH, ISDELETE)** – nhân bản toàn bộ

- **NHANVIEN_PRIVATE(MANV, LUONG, MACHINHANH, ISDELETE)** –
phân mảnh theo chi nhánh

HOADON(MAHD, MANV, MAKH, NGAYTAO, TONGTIEN, ISDELETE)

Tân từ: Hóa đơn gồm mã hóa đơn (MAHD), nhân viên thanh toán (MANV), khách hàng (MAKH), ngày tạo hóa đơn (NGAYTAO), tổng tiền (TONGTIEN). Bảng này được phân mảnh ngang theo chi nhánh. ISDELETE dùng để xoá mềm hoá đơn thay vì xoá vĩnh viễn.

CTHD(MAHD, MASP, SOLUONG, ISDELETE)

Tân từ: Chi tiết hóa đơn gồm mã hóa đơn (MAHD), mã sản phẩm (MASP), số lượng mua (SOLUONG). Phân mảnh cùng với bảng HOADON. ISDELETE để xoá mềm dòng dữ liệu của chi tiết hóa đơn.

TONKHO(MASP, MACHINHANH, SOLUONG, TINHTRANG, NGAYCAPNHAT)

Tân từ: Thông tin số lượng hàng tồn của sản phẩm tại mỗi chi nhánh. Bảng này phân mảnh theo chi nhánh.

1.3. Phân mảnh dữ liệu

1.3.1. Phân mảnh ngang

HOADON / CTHD:

- CN1: HOADON, CTHD, TONKHO, NHANVIEN_PUBLIC, NHANVIEN_PRIVATE
- CN2: HOADON, CTHD, TONKHO, NHANVIEN_PUBLIC, NHANVIEN_PRIVATE
- CN3: HOADON, CTHD, TONKHO, NHANVIEN_PUBLIC, NHANVIEN_PRIVATE

1.3.2. Phân mảnh dọc

NHANVIEN:

- NHANVIEN_PUBLIC(MANV, HOTEN, GIOITINH, NGAYSINH, DIACHI, SODIENTHOAI, NGAYVAOLAM, CHUCVU MACHINHANH, ISDELETE) –
phân mảnh theo chi nhánh

- NHANVIEN_PRIVATE(MANV, LUONG, MACHINHANH) – phân mảnh theo chi nhánh

1.3.3. Nhân bản (Replicated)

- SANPHAM – replicated toàn bộ tại mỗi chi nhánh
- KHACHHANG – replicated toàn bộ tại mỗi chi nhánh

1.4. Phân bố dữ liệu theo máy ảo

Bảng 1. Phân bố dữ liệu

Máy	Chi nhánh	Dữ liệu
Hải Hà	CN1 (TP.HCM)	HOADON, CTHD, TONKHO NHANVIEN_PRIVATE, NHANVIEN_PUBLIC
Nhật Long	CN2 (Hà Nội)	HOADON, CTHD, TONKHO, NHANVIEN_PRIVATE, NHANVIEN_PUBLIC
Đức Mạnh	CN3 (Đà Nẵng)	HOADON, CTHD, TONKHO, NHANVIEN_PRIVATE, NHANVIEN_PUBLIC
3 máy	Share	SANPHAM, KHACHHANG

1.4.1. Quyền truy cập đề xuất

CN1:

- **GiamDoc:** SELECT, INSERT, UPDATE tất cả bảng ở 3 chi nhánh.
- **QuanLy1:** SELECT, INSERT, UPDATE SANPHAM, TONKHO, HOADON, CTHD, KHACHHANG; SELECT NHANVIEN_PUBLIC, NHANVIEN_PRIVATE chi nhánh 1; SELECT NHANVIEN_PUBLIC, TONKHO chi nhánh 2, 3.
- **NhanVien1:** SELECT, INSERT, UPDATE HOADON, CTHD, KHACHHANG của chi nhánh 1; SELECT SANPHAM của chi nhánh 1; SELECT TONKHO chi nhánh 2, 3.

CN2 & CN3: giống CN1 nhưng trên dữ liệu chi nhánh tương ứng.

Bảng 2. Tạo bảng

```
-- BẢNG CHI NHÁNH
CREATE TABLE CHINHANH (
    MACHINHANH VARCHAR2(10) PRIMARY KEY,
```

```
TENCHINHANH VARCHAR2(100),
DIACHI VARCHAR2(200),
SODIENTHOAI VARCHAR2(20)
);

-- BẢNG NHÂN VIÊN CÔNG KHAI (Public Info)
CREATE TABLE NHANVIEN_PUBLIC (
    MANV VARCHAR2(10) PRIMARY KEY,
    HOTEN VARCHAR2(100),
    GIOITINH VARCHAR2(10),
    NGAYSINH DATE,
    DIACHI VARCHAR2(200),
    SODIENTHOAI VARCHAR2(15),
    NGÀYVAOLAM DATE,
    CHUCVU VARCHAR2(50),
    MACHINHANH VARCHAR2(10),
    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1)),
    CONSTRAINT FK_NV_PUBLIC_CN FOREIGN KEY (MACHINHANH) REFERENCES
    CHINHANH(MACHINHANH)
);

-- BẢNG NHÂN VIÊN RIÊNG (Private Info)
CREATE TABLE NHANVIEN_PRIVATE (
    MANV VARCHAR2(10) PRIMARY KEY,
    LUONG NUMBER,
    MACHINHANH VARCHAR2(10),
    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1)),
    CONSTRAINT FK_NV_PRIVATE_PUBLIC FOREIGN KEY (MANV) REFERENCES
    NHANVIEN_PUBLIC(MANV)
);

-- BẢNG KHÁCH HÀNG
CREATE TABLE KHACHHANG (
    MAKH VARCHAR2(10) PRIMARY KEY,
    HOTEN VARCHAR2(100),
    DIACHI VARCHAR2(200),
    SODIENTHOAI VARCHAR2(20),
    TONGTIEN NUMBER DEFAULT 0,
    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1))
);

-- BẢNG SẢN PHẨM
CREATE TABLE SANPHAM (
    MASP VARCHAR2(10) PRIMARY KEY,
    TENSANPHAM VARCHAR2(200),
    DANHMUC VARCHAR2(50),
    THUONGHIEU VARCHAR2(50),
    GIANHAP NUMBER,
    GIABAN NUMBER,
```

```

    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1))
);

-- BẢNG TỒN KHO
CREATE TABLE TONKHO (
    MASP VARCHAR2(10),
    MACHINHANH VARCHAR2(10),
    SOLUONG NUMBER,
    TINHTRANG VARCHAR2(20), -- 'Còn Hàng' hoặc 'Hết Hàng'
    NGAYCAPNHAT DATE,
    PRIMARY KEY (MASP, MACHINHANH),
    CONSTRAINT FK_TK_SP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP),
    CONSTRAINT FK_TK_CN FOREIGN KEY (MACHINHANH) REFERENCES
CHINHANH(MACHINHANH)
);

-- BẢNG HÓA ĐƠN
CREATE TABLE HOADON (
    MAHD VARCHAR2(10) PRIMARY KEY,
    MANV VARCHAR2(10),
    MAKH VARCHAR2(10),
    NGAYTAO DATE,
    TONGTIEN NUMBER,
    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1)),
    CONSTRAINT FK_HD_NV FOREIGN KEY (MANV) REFERENCES NHANVIEN_PUBLIC(MANV),
    CONSTRAINT FK_HD_KH FOREIGN KEY (MAKH) REFERENCES KHACHHANG(MAKH)
);

-- BẢNG CHI TIẾT HÓA ĐƠN
CREATE TABLE CTHD (
    MAHD VARCHAR2(10),
    MASP VARCHAR2(10),
    SOLUONG NUMBER,
    ISDELETE NUMBER(1) DEFAULT 0 CHECK (ISDELETE IN (0,1)),
    PRIMARY KEY (MAHD, MASP),
    CONSTRAINT FK_CTHD_HD FOREIGN KEY (MAHD) REFERENCES HOADON(MAHD),
    CONSTRAINT FK_CTHD_SP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP)
);

-- Thêm dữ liệu vào bảng CHINHANH
INSERT INTO CHINHANH (MACHINHANH, TENCHINHANH, DIACHI, SODIENTHOAI) VALUES
('CN1', 'Chi nhánh TP.HCM', '123 Lê Lợi, Quận 1, TP.HCM', '02812345678');

INSERT INTO CHINHANH (MACHINHANH, TENCHINHANH, DIACHI, SODIENTHOAI) VALUES
('CN2', 'Chi nhánh Hà Nội', '456 Trần Duy Hưng, Cầu Giấy, Hà Nội',
'02412345678');

INSERT INTO CHINHANH (MACHINHANH, TENCHINHANH, DIACHI, SODIENTHOAI) VALUES

```

```
('CN3', 'Chi nhánh Đà Nẵng', '789 Nguyễn Văn Linh, Hải Châu, Đà Nẵng',
'02361234567');
```

1.4.2. Phân quyền: (tự thêm số vô TRUONGCN với NHANVIEN nha)

Bảng 3. Phân quyền

```
-- TẠO NGƯỜI DÙNG
CREATE USER GiamDoc IDENTIFIED BY giamdoc;
CREATE USER QuanLy1 IDENTIFIED BY quanly1;
CREATE USER NhanVien1 IDENTIFIED BY nhanvien1;
CREATE USER QuanLy2 IDENTIFIED BY quanly2;
CREATE USER QuanLy3 IDENTIFIED BY quanly3;
CREATE USER NhanVien2 IDENTIFIED BY nhanvien2;
CREATE USER NhanVien3 IDENTIFIED BY nhanvien3;

-- CẤP QUYỀN KẾT NỐI CHO TẤT CẢ USER
GRANT CONNECT TO GiamDoc, QuanLy1, NhanVien1;
GRANT CONNECT TO QuanLy2, QuanLy3, NhanVien2, NhanVien3;

-- PHÂN QUYỀN CHO GIAMDOC: SELECT, INSERT, UPDATE TẤT CẢ CÁC BẢNG
GRANT SELECT, INSERT, UPDATE ON SANPHAM TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON TONKHO TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON HOADON TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON CTHD TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON KHACHHANG TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON NHANVIEN_PUBLIC TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON NHANVIEN_PRIVATE TO GiamDoc;
GRANT SELECT, INSERT, UPDATE ON CHINHANH TO GiamDoc;

-- PHÂN QUYỀN CHO QUANLY1: SELECT, INSERT, UPDATE SANPHAM, TONKHO, HOADON,
CTHD, KHACHHANG; SELECT NHANVIEN_PUBLIC, NHANVIEN_PRIVATE; SELECT
NHANVIEN_PUBLIC và NHANVIEN_PRIVATE
GRANT SELECT, INSERT, UPDATE ON SANPHAM TO QuanLy1;
GRANT SELECT, INSERT, UPDATE ON TONKHO TO QuanLy1;
GRANT SELECT, INSERT, UPDATE ON HOADON TO QuanLy1;
GRANT SELECT, INSERT, UPDATE ON CTHD TO QuanLy1;
GRANT SELECT, INSERT, UPDATE ON KHACHHANG TO QuanLy1;
GRANT SELECT ON NHANVIEN_PUBLIC TO QuanLy1;
GRANT SELECT ON NHANVIEN_PRIVATE TO QuanLy1;

-- PHÂN QUYỀN CHO NHANVIEN: SELECT, INSERT, UPDATE KHACHHANG, HOADON, CTHD;
SELECT SANPHAM, TONKHO
GRANT SELECT, INSERT, UPDATE ON KHACHHANG TO NhanVien1;
GRANT SELECT, INSERT, UPDATE ON HOADON TO NhanVien1;
GRANT SELECT, INSERT, UPDATE ON CTHD TO NhanVien1;
GRANT SELECT ON SANPHAM TO NhanVien1;
GRANT SELECT ON TONKHO TO NhanVien1;
```

```
-- PHÂN QUYỀN CHO QUẢN LÝ CHI NHÁNH 2, 3: SELECT NHANVIEN_PUBLIC, TONKHO
GRANT SELECT ON NHANVIEN_PUBLIC TO QuanLy2, QuanLy3;
GRANT SELECT ON TONKHO TO QuanLy2, QuanLy3;

-- PHÂN QUYỀN CHO NHÂN VIÊN CHI NHÁNH 2, 3: SELECT TONKHO
GRANT SELECT ON TONKHO TO NhanVien2, NhanVien3;
```

1.4.3. Tạo Database link

Bảng 4. Tạo Database link

```
-- CN1
CREATE PUBLIC DATABASE LINK CN2_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'long_link';
CREATE PUBLIC DATABASE LINK CN2_QuanLy1 CONNECT TO QuanLy2 IDENTIFIED BY
quanly2 USING 'long_link';
CREATE PUBLIC DATABASE LINK CN2_NhanVien1 CONNECT TO NhanVien2 IDENTIFIED BY
nhanvien2 USING 'long_link';

CREATE PUBLIC DATABASE LINK CN3_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'pn1_link';
CREATE PUBLIC DATABASE LINK CN3_QuanLy1 CONNECT TO QuanLy1 IDENTIFIED BY
quanly1 USING 'pn1_link';
CREATE PUBLIC DATABASE LINK CN3_NhanVien1 CONNECT TO NhanVien3 IDENTIFIED BY
nhanvien3 USING 'pn1_link';

-- CN2
CREATE PUBLIC DATABASE LINK CN1_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'haiha_link';
CREATE PUBLIC DATABASE LINK CN1_QuanLy2 CONNECT TO QuanLy2 IDENTIFIED BY
quanly2 USING 'haiha _link';
CREATE PUBLIC DATABASE LINK CN1_NhanVien2 CONNECT TO NhanVien2 IDENTIFIED BY
nhanvien2 USING 'haiha_link';

CREATE PUBLIC DATABASE LINK CN3_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'manh_link';
CREATE PUBLIC DATABASE LINK CN3_QuanLy2 CONNECT TO QuanLy2 IDENTIFIED BY
quanly2 USING 'manh_link';
CREATE PUBLIC DATABASE LINK CN3_NhanVien2 CONNECT TO NhanVien2 IDENTIFIED BY
nhanvien2 USING 'manh_link';

-- CN3
CREATE PUBLIC DATABASE LINK CN1_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'haiha_link';
CREATE PUBLIC DATABASE LINK CN1_QuanLy3 CONNECT TO QuanLy3 IDENTIFIED BY
quanly3 USING 'haiha_link';
CREATE PUBLIC DATABASE LINK CN1_NhanVien3 CONNECT TO NhanVien3 IDENTIFIED BY
nhanvien3 USING 'haiha_link';
```



```
CREATE PUBLIC DATABASE LINK CN2_GiamDoc CONNECT TO GiamDoc IDENTIFIED BY
giamdoc USING 'long_link';
CREATE PUBLIC DATABASE LINK CN2_QuanLy3 CONNECT TO QuanLy3 IDENTIFIED BY
quanly3 USING 'long_link';
CREATE PUBLIC DATABASE LINK CN2_NhanVien3 CONNECT TO NhanVien3 IDENTIFIED BY
nhanvien3 USING 'long_link';
```

1.5. Câu truy vấn

1.5.1. (Thực hiện ở CN1) Giám đốc tìm nhân viên xuất sắc nhất năm 2024 của mỗi chi nhánh (doanh thu cao nhất)

```
SELECT MACHINHANH, MANV, HOTEN, DOANHTHUCANHAN
FROM (
    SELECT MACHINHANH, N.MANV, N.HOTEN, SUM(H.TONGTIEN) AS DOANHTHUCANHAN
    FROM CN1.NHANVIEN_PUBLIC N
    JOIN CN1.HOADON H ON N.MANV = H.MANV
    WHERE H.ISDELETE = 0 AND EXTRACT(YEAR FROM H.NGAYTAO) = 2024
    GROUP BY N.MACHINHANH, N.MANV, N.HOTEN
    ORDER BY SUM(H.TONGTIEN) DESC
    FETCH FIRST 1 ROW WITH TIES
)
UNION ALL
SELECT MACHINHANH, MANV, HOTEN, DOANHTHUCANHAN
FROM (
    SELECT MACHINHANH, N.MANV, N.HOTEN, SUM(H.TONGTIEN) AS DOANHTHUCANHAN
    FROM CN2.NHANVIEN_PUBLIC@CN2_GiamDoc N
    JOIN CN2.HOADON@CN2_GiamDoc H ON N.MANV = H.MANV
    WHERE H.ISDELETE = 0 AND EXTRACT(YEAR FROM H.NGAYTAO) = 2024
    GROUP BY N.MACHINHANH, N.MANV, N.HOTEN
    ORDER BY SUM(H.TONGTIEN) DESC
    FETCH FIRST 1 ROW WITH TIES
)
UNION ALL
SELECT MACHINHANH, MANV, HOTEN, DOANHTHUCANHAN
FROM (
    SELECT MACHINHANH, N.MANV, N.HOTEN, SUM(H.TONGTIEN) AS DOANHTHUCANHAN
    FROM CN3.NHANVIEN_PUBLIC@CN3_GiamDoc N
    JOIN CN3.HOADON@CN3_GiamDoc H ON N.MANV = H.MANV
    WHERE H.ISDELETE = 0 AND EXTRACT(YEAR FROM H.NGAYTAO) = 2024
    GROUP BY N.MACHINHANH, N.MANV, N.HOTEN
    ORDER BY SUM(H.TONGTIEN) DESC
    FETCH FIRST 1 ROW WITH TIES
);
```

Kết quả

MACHINHANH VARCHAR2(10)	MANV VARCHAR2(10)	HOTEN VARCHAR2(100)	DOANHTHUCANHAN NUMBER
CN1	NV0239	Đặng Văn Nam	7133141000
CN2	NV0401	Nguyễn Thị Yến	6824077000
CN3	NV0916	Bùi Văn Phong	7114153500

Hình 1-1. Kết quả câu truy vấn số 1

1.5.2. (Thực hiện ở CN1) Giám đốc thống kê doanh thu theo từng chi nhánh trong năm 2024, sau đó tính tổng doanh thu của hệ thống.

```

SELECT
    NVL(MACHINHANH, 'TONG') AS MACHINHANH,
    SUM(TONGTIEN) AS DOANHTHU
FROM (
    SELECT HD.MANV, NV.MACHINHANH, HD.TONGTIEN
    FROM CN1.HOADON HD
    JOIN CN1.NHANVIEN_PUBLIC NV ON HD.MANV = NV.MANV
    WHERE HD.ISDELETE = 0 AND EXTRACT(YEAR FROM HD.NGAYTAO) = 2024
    UNION ALL
    SELECT HD.MANV, NV.MACHINHANH, HD.TONGTIEN
    FROM CN2.HOADON@CN2_GiamDoc HD
    JOIN CN2.NHANVIEN_PUBLIC@CN2_GiamDoc NV ON HD.MANV = NV.MANV
    WHERE HD.ISDELETE = 0 AND EXTRACT(YEAR FROM HD.NGAYTAO) = 2024
    UNION ALL
    SELECT HD.MANV, NV.MACHINHANH, HD.TONGTIEN
    FROM CN3.HOADON@CN3_GiamDoc HD
    JOIN CN3.NHANVIEN_PUBLIC@CN3_GiamDoc NV ON HD.MANV = NV.MANV
    WHERE HD.ISDELETE = 0 AND EXTRACT(YEAR FROM HD.NGAYTAO) = 2024
)
GROUP BY ROLLUP(MACHINHANH)

```

Kết quả

MACHINHANH VARCHAR2(4)	DOANHTHU NUMBER
CN1	1766842783000
CN2	1757670699500
CN3	1789127333000
TONG	5313640815500

Hình 1-2. Kết quả câu truy vấn số 2

1.5.3. (Thực hiện ở CN1) Giám đốc tìm ra khách hàng tiềm năng đã mua hàng từ 2 chi nhánh trở lên

```

SELECT MAKH, MASP, (SELECT TENSANPHAM FROM CN1.SANPHAM WHERE MASP = DS.MASP)
AS TENSANPHAM
FROM (
    SELECT H.MAKH, C.MASP, NV.MACHINHANH
    FROM CN1.HOADON H
    JOIN CN1.CTHD C ON H.MAHD = C.MAHD
    JOIN CN1.NHANVIEN_PUBLIC NV ON H.MANV = NV.MANV
    WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
    UNION ALL
    SELECT H.MAKH, C.MASP, NV.MACHINHANH
    FROM CN2.HOADON@CN2_GiamDoc H
    JOIN CN2.CTHD@CN2_GiamDoc C ON H.MAHD = C.MAHD
    JOIN CN2.NHANVIEN_PUBLIC@CN2_GiamDoc NV ON H.MANV = NV.MANV
    WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
    UNION ALL
    SELECT H.MAKH, C.MASP, NV.MACHINHANH
    FROM CN3.HOADON@CN3_GiamDoc H
    JOIN CN3.CTHD@CN3_GiamDoc C ON H.MAHD = C.MAHD
    JOIN CN3.NHANVIEN_PUBLIC@CN3_GiamDoc NV ON H.MANV = NV.MANV
    WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
) DS
GROUP BY MAKH, MASP
HAVING COUNT(DISTINCT MACHINHANH) >= 2;

```

Kết quả

MAKH VARCHAR2(10)	MASP VARCHAR2(10)	MACHINHANH VARCHAR2(10)	TENSANPHAM VARCHAR2(200)
KH0001	SP0006	CN1	Vợt cầu lông Yonex Nanoflare 700 Mag
KH0001	SP0006	CN3	Vợt cầu lông Yonex Nanoflare 700 Mag

Hình 1-3. Kết quả câu truy vấn số 3

1.5.4. (Thực hiện ở CN1) Giám đốc thống kê mức lương trung bình trả cho một nhân viên tại mỗi chi nhánh.

```

SELECT MACHINHANH, ROUND(AVG(LUONG), 0) AS LUONG_TRUNG_BINH
FROM (
    SELECT MACHINHANH, LUONG
    FROM CN1.NHANVIEN_PRIVATE
    WHERE ISDELETE = 0
    UNION ALL
    SELECT MACHINHANH, LUONG
    FROM CN2.NHANVIEN_PRIVATE@CN2_GiamDoc
    WHERE ISDELETE = 0
    UNION ALL
    SELECT MACHINHANH, LUONG

```

```

FROM CN3.NHANVIEN_PRIVATE@CN3_GiamDoc
WHERE ISDELETE = 0
)
GROUP BY MACHINHANH
ORDER BY MACHINHANH;

```

Kết quả

MACHINHANH VARCHAR2(10)	LUONG_TRUNG_BINH NUMBER
CN1	6978360
CN2	7021046
CN3	7041848

Hình 1-4. Kết quả câu truy vấn số 4

1.5.5. (Thực hiện ở CN2) Nhân viên tìm các sản phẩm còn hàng thuộc danh mục 'Vợt câu lông' có giá trên 2 triệu dưới 3 triệu.

```

SELECT
  MASP,
  TENSANPHAM,
  GIABAN,
  SUM(CASE WHEN MACHINHANH = 'CN1' THEN SOLUONG ELSE 0 END) AS CN1,
  SUM(CASE WHEN MACHINHANH = 'CN2' THEN SOLUONG ELSE 0 END) AS CN2,
  SUM(CASE WHEN MACHINHANH = 'CN3' THEN SOLUONG ELSE 0 END) AS CN3,
  SUM(SOLUONG) AS TONG
FROM (
  SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
  FROM CN1.TONKHO@CN1_NhanVien2 T
  JOIN CN1.SANPHAM@CN1_NhanVien2 S ON T.MASP = S.MASP
  WHERE T.TINHTRANG = 'Còn Hàng'
    AND S.DANH MUC = 'Vợt câu lông'
    AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
    AND S.ISDELETE = 0
  UNION ALL
  SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
  FROM CN2.TONKHO T
  JOIN CN2.SANPHAM S ON T.MASP = S.MASP
  WHERE T.TINHTRANG = 'Còn Hàng'
    AND S.DANH MUC = 'Vợt câu lông'
    AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
    AND S.ISDELETE = 0
  UNION ALL
  SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
  FROM CN3.TONKHO@CN3_NhanVien2 T
  JOIN CN3.SANPHAM@CN3_NhanVien2 S ON T.MASP = S.MASP
  WHERE T.TINHTRANG = 'Còn Hàng'
    AND S.DANH MUC = 'Vợt câu lông'

```

```

        AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
        AND S.ISDELETE = 0
    )
GROUP BY MASP, TENSANPHAM, GIABAN
ORDER BY TENSANPHAM;
SELECT
    MASP,
    TENSANPHAM,
    GIABAN,
    SUM(CASE WHEN MACHINHANH = 'CN1' THEN SOLUONG ELSE 0 END) AS CN1,
    SUM(CASE WHEN MACHINHANH = 'CN2' THEN SOLUONG ELSE 0 END) AS CN2,
    SUM(CASE WHEN MACHINHANH = 'CN3' THEN SOLUONG ELSE 0 END) AS CN3,
    SUM(SOLUONG) AS TONG
FROM (
    SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
    FROM CN1.TONKHO@CN1_NhanVien2 T
    JOIN CN1.SANPHAM@CN1_NhanVien2 S ON T.MASP = S.MASP
    WHERE T.TINHTRANG = 'Còn Hàng'
        AND S.DANHMUC = 'Vật cầu lông'
        AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
        AND S.ISDELETE = 0
    UNION ALL
    SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
    FROM CN2.TONKHO T
    JOIN CN2.SANPHAM S ON T.MASP = S.MASP
    WHERE T.TINHTRANG = 'Còn Hàng'
        AND S.DANHMUC = 'Vật cầu lông'
        AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
        AND S.ISDELETE = 0
    UNION ALL
    SELECT T.MASP, S.TENSANPHAM, S.GIABAN, T.MACHINHANH, T.SOLUONG
    FROM CN3.TONKHO@CN3_NhanVien2 T
    JOIN CN3.SANPHAM@CN3_NhanVien2 S ON T.MASP = S.MASP
    WHERE T.TINHTRANG = 'Còn Hàng'
        AND S.DANHMUC = 'Vật cầu lông'
        AND S.GIABAN > 2000000 AND S.GIABAN < 3000000
        AND S.ISDELETE = 0
    )
GROUP BY MASP, TENSANPHAM, GIABAN
ORDER BY TENSANPHAM;

```

Kết quả

MASP VARCHAR2(10)	TENSANPHAM VARCHAR2(200)	GIABAN NUMBER	CN1 NUMBER	CN2 NUMBER	CN3 NUMBER	TONG NUMBER
SP0743	Combo mua vợt cầu lông Victor ARS 30H tặng vợt Victor HX063 + vợt Victor TK110	2430000	134	122	147	403
SP0940	Combo mua vợt cầu lông Victor BRS 1900 tặng vợt Victor BRS 1900	2030000	200	189	108	497
SP0908	Combo mua vợt cầu lông Victor Hypermano X080 tặng vợt Victor Hypermano X080 + Áo Suves Victor 5362DC/ Tặng vợt Victor Hypermano X08...	2280000	105	106	149	360
SP0687	Set vợt cầu lông Victor Doraemon TK-DRM	2349000	64	0	54	118
SP0762	Vợt cầu lông Victor Hypermano X090 chính hãng	2300000	131	145	38	314
SP0485	Vợt Cầu Lông Lining Axforce BigBang	2100000	45	114	198	357
SP0497	Vợt Cầu Lông Lining Calbar 600	2750000	158	96	146	400
SP0518	Vợt Cầu Lông Lining Calbar 600B	2750000	165	37	176	378
SP0620	Vợt Cầu Lông Lining Calbar 600C	2750000	8	178	46	232
SP0486	Vợt Cầu Lông Lining Calbar 600i	2750000	127	172	126	425
SP0546	Vợt Cầu Lông Lining FLAME F360 chính hãng	2500000	34	43	116	193
SP0576	Vợt Cầu Lông Lining Flame N55	2890000	4	169	174	347
SP0547	Vợt Cầu Lông Lining TB Nano 120 chính hãng	2740000	108	173	93	374
SP0520	Vợt Cầu Lông Lining Technic 6	2300000	171	71	167	409

Hình 1-5. Kết quả câu truy vấn số 5

1.5.6. (Thực hiện ở CN2) Giám đốc tìm khách hàng đã mua tất cả sản phẩm từ hãng VNB

```

SELECT KH.MAKH, KH.HOTEN, SP.MASP, SP.TENSANPHAM
FROM CN2.KHACHHANG KH
JOIN (
    SELECT MAKH
    FROM (
        SELECT H.MAKH, C.MASP
        FROM CN1.HOADON@CN1_GiamDoc H
        JOIN CN1.CTHD@CN1_GiamDoc C ON H.MAHD = C.MAHD
        WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
        AND C.MASP IN (SELECT MASP FROM CN2.SANPHAM WHERE THUONGHIEU = 'VNB'
AND ISDELETE = 0)
        UNION
        SELECT H.MAKH, C.MASP
        FROM CN2.HOADON H
        JOIN CN2.CTHD C ON H.MAHD = C.MAHD
        WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
        AND C.MASP IN (SELECT MASP FROM CN2.SANPHAM WHERE THUONGHIEU = 'VNB'
AND ISDELETE = 0)
        UNION
        SELECT H.MAKH, C.MASP
        FROM CN3.HOADON@CN3_GiamDoc H
        JOIN CN3.CTHD@CN3_GiamDoc C ON H.MAHD = C.MAHD
        WHERE H.ISDELETE = 0 AND C.ISDELETE = 0
        AND C.MASP IN (SELECT MASP FROM CN2.SANPHAM WHERE THUONGHIEU = 'VNB'
AND ISDELETE = 0)
    )
    GROUP BY MAKH
    HAVING COUNT(DISTINCT MASP) = (SELECT COUNT(*) FROM CN2.SANPHAM WHERE
THUONGHIEU = 'VNB' AND ISDELETE = 0)
) KD ON KH.MAKH = KD.MAKH
JOIN CN2.SANPHAM SP ON SP.THUONGHIEU = 'VNB' AND SP.ISDELETE = 0
ORDER BY KH.MAKH, SP.MASP;

```

Kết quả

1.5.7. (Thực hiện ở CN2) Giám đốc tìm ra chi nhánh có mức tăng trưởng doanh thu tốt nhất năm 2024 (doanh thu cả năm so với doanh thu 6 tháng đầu tiên)

```

WITH DOANHTHU AS (
    SELECT 'CN1' AS MACHINHANH, 'FULL' AS KY, SUM(TONGTIEN) AS DOANHTHU
    FROM CN1.HOADON@CN1_GiamDoc
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND ISDELETE = 0
    UNION ALL
    SELECT 'CN2' AS MACHINHANH, 'FULL', SUM(TONGTIEN)
    FROM CN2.HOADON
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND ISDELETE = 0
    UNION ALL
    SELECT 'CN3' AS MACHINHANH, 'FULL', SUM(TONGTIEN)
    FROM CN3.HOADON@CN3_GiamDoc
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND ISDELETE = 0

    UNION ALL

    SELECT 'CN1' AS MACHINHANH, 'FIRST6', SUM(TONGTIEN)
    FROM CN1.HOADON@CN1_GiamDoc
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND EXTRACT(MONTH FROM NGAYTAO)
    BETWEEN 1 AND 6 AND ISDELETE = 0
    UNION ALL
    SELECT 'CN2' AS MACHINHANH, 'FIRST6', SUM(TONGTIEN)
    FROM CN2.HOADON
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND EXTRACT(MONTH FROM NGAYTAO)
    BETWEEN 1 AND 6 AND ISDELETE = 0
    UNION ALL
    SELECT 'CN3' AS MACHINHANH, 'FIRST6', SUM(TONGTIEN)
    FROM CN3.HOADON@CN3_GiamDoc
    WHERE EXTRACT(YEAR FROM NGAYTAO) = 2024 AND EXTRACT(MONTH FROM NGAYTAO)
    BETWEEN 1 AND 6 AND ISDELETE = 0
)

SELECT MACHINHANH,
       MAX(CASE WHEN KY = 'FULL' THEN DOANHTHU ELSE 0 END) AS DOANHTHU_NAM,
       MAX(CASE WHEN KY = 'FIRST6' THEN DOANHTHU ELSE 0 END) AS
DOANHTHU_6THANG,
       MAX(CASE WHEN KY = 'FULL' THEN DOANHTHU ELSE 0 END) -
       MAX(CASE WHEN KY = 'FIRST6' THEN DOANHTHU ELSE 0 END) AS TANGTRUONG
FROM DOANHTHU
GROUP BY MACHINHANH
ORDER BY TANGTRUONG DESC
FETCH FIRST 1 ROW ONLY;

```

Kết quả

MACHINHANH CHAR(3)	DOANHTHU_NAM NUMBER	DOANHTHU_6THANG NUMBER	TANGTRUONG NUMBER
CN3	1789127333000	884632887500	904494445500

Hình 1-6. Kết quả câu truy vấn số 7

1.5.8. (Thực hiện ở CN3) Quản lý liệt kê số lượng các nhân viên làm việc tại từng chi nhánh

```

SELECT MACHINHANH, SOLUONGNHANVIEN
FROM (
SELECT MACHINHANH, COUNT(MANV) AS SOLUONGNHANVIEN
FROM CN1.NHANVIEN_PUBLIC@CN1_QuanLy3
WHERE ISDELETE = 0 AND MACHINHANH = 'CN1'
GROUP BY MACHINHANH
UNION ALL
SELECT MACHINHANH, COUNT(MANV) AS SOLUONGNHANVIEN
FROM CN2.NHANVIEN_PUBLIC@CN2_QuanLy3
WHERE ISDELETE = 0 AND MACHINHANH = 'CN2'
GROUP BY MACHINHANH
UNION ALL
SELECT MACHINHANH, COUNT(MANV) AS SOLUONGNHANVIEN
FROM CN3.NHANVIEN_PUBLIC
WHERE ISDELETE = 0 AND MACHINHANH = 'CN3'
GROUP BY MACHINHANH
);

```

Kết quả

	MACHINHANH VARCHAR2(3)	SOLUONGNHANVIEN NUMBER
▶	CN1	334
	CN2	333
	CN3	333

Hình 1-7. Kết quả câu truy vấn số 8

1.5.9. (Thực hiện ở CN3) Nhân viên tìm sản phẩm hết hàng ở CN3 nhưng còn hàng ở CN1 và CN2.

```

SELECT MASP, TENSANPHAM
FROM (
SELECT T.MASP, S.TENSANPHAM
FROM CN1.TONKHO@CN1_NhanVien3 T
JOIN CN1.SANPHAM@CN1_NhanVien3 S ON T.MASP = S.MASP
WHERE T.TINHTRANG = 'Còn Hàng' AND S.ISDELETE = 0
INTERSECT
SELECT T.MASP, S.TENSANPHAM
FROM CN2.TONKHO@CN2_NhanVien3 T

```

```

JOIN CN2.SANPHAM@CN2_NhanVien3 S ON T.MASP = S.MASP
WHERE T.TINHTRANG = 'Còn Hàng' AND S.ISDELETE = 0
)
MINUS
SELECT T.MASP, S.TENSANPHAM
FROM CN3.TONKHO T
JOIN CN3.SANPHAM S ON T.MASP = S.MASP
WHERE T.TINHTRANG = 'Còn Hàng' AND S.ISDELETE = 0
GROUP BY T.MASP, S.TENSANPHAM

```

Kết quả

	MASP VARCHAR2(10)	TENSANPHAM VARCHAR2(200)
▶	SP0887	Vợt cầu lông Victor ARS-77F
	SP0093	Vợt cầu lông Yonex Astrox 88D Tour chính hãng
	SP0415	Vợt Cầu Lông Yonex Duora 6 chính hãng
	SP0652	Vợt cầu lông Lining Tectonic 9
	SP0829	Vợt cầu lông Victor Hypernano X60 chính hãng
	SP0050	Vợt cầu lông Yonex NanoFlare 800
	SP0385	Vợt cầu lông Yonex Voltric 0.5 DG Slim chính hãng
	SP0793	Vợt cầu lông Victor Hypernano DF09 chính hãng
	SP0435	Vợt cầu lông Yonex Duora 8XP

Hình 1-8. Kết quả câu truy vấn số 9

1.5.10. (Thực hiện tại CN3) Giám đốc thống kê Top 5 sản phẩm bán chạy nhất toàn hệ thống

```

SELECT MASP, TENSANPHAM, SUM(SOLUONG) AS TONG_SOLUONG
FROM (
  SELECT C.MASP, S.TENSANPHAM, C.SOLUONG
  FROM CN1.CTHD@CN1_GiamDoc C
  JOIN CN1.SANPHAM@CN1_GiamDoc S ON C.MASP = S.MASP
  WHERE C.ISDELETE = 0 AND S.ISDELETE = 0
  UNION ALL
  SELECT C.MASP, S.TENSANPHAM, C.SOLUONG
  FROM CN2.CTHD@CN2_GiamDoc C
  JOIN CN2.SANPHAM@CN2_GiamDoc S ON C.MASP = S.MASP
  WHERE C.ISDELETE = 0 AND S.ISDELETE = 0
  UNION ALL
  SELECT C.MASP, S.TENSANPHAM, C.SOLUONG
  FROM CN3.CTHD C
  JOIN CN3.SANPHAM S ON C.MASP = S.MASP
  WHERE C.ISDELETE = 0 AND S.ISDELETE = 0
)
GROUP BY MASP, TENSANPHAM

```

```
ORDER BY TONG_SOLUONG DESC
FETCH FIRST 5 ROWS WITH TIES;
```

Kết quả

MAASP VARCHAR2(10)	TENSANPHAM VARCHAR2(200)	TONG_SOLUONG NUMBER
SP0013	Vợt cầu lông Yonex NanoFlare 600 chính hãng	12809
SP0372	Vợt Cầu Lông Yonex Voltric D20 chính hãng	12738
SP0965	Vợt Cầu Lông Victor BS 1700 N chính hãng	12708
SP1008	Quần cán Baloli	12681
SP1173	Dây Cước Căng Vợt Victor Hello Kitty VS-KT63	12672

Hình 1-9. Kết quả câu truy vấn số 10

2. VIẾT HÀM, THỦ TỤC VÀ RÀNG BUỘC TOÀN VỆ

2.1. Function

2.1.1. Function thống kê mua hàng khách hàng trên toàn hệ thống

```
CREATE OR REPLACE FUNCTION ThongKeMuaHangKhachHang(p_makh IN VARCHAR2)
RETURN NUMBER
IS
    v_count_cn1 NUMBER := 0;
    v_sum_cn1    NUMBER := 0;
    v_count_cn2 NUMBER := 0;
    v_sum_cn2    NUMBER := 0;
    v_count_cn3 NUMBER := 0;
    v_sum_cn3    NUMBER := 0;
    v_total_hd   NUMBER := 0;
    v_total_tien NUMBER := 0;
BEGIN
    -- CN1
    BEGIN
        SELECT COUNT(*), NVL(SUM(TONGTIEN), 0)
        INTO v_count_cn1, v_sum_cn1
        FROM CN1.HOADON@CN1_GiamDoc
        WHERE MAKH = p_makh AND ISDELETE = 0;
    EXCEPTION WHEN OTHERS THEN
        v_count_cn1 := 0;
        v_sum_cn1 := 0;
    END;

    -- CN2 (local)
```

```

BEGIN
    SELECT COUNT(*), NVL(SUM(TONGTIEN), 0)
    INTO v_count_cn2, v_sum_cn2
    FROM HOADON
    WHERE MAKH = p_makh AND ISDELETE = 0;
EXCEPTION WHEN OTHERS THEN
    v_count_cn2 := 0;
    v_sum_cn2 := 0;
END;

-- CN3
BEGIN
    SELECT COUNT(*), NVL(SUM(TONGTIEN), 0)
    INTO v_count_cn3, v_sum_cn3
    FROM CN3.HOADON@CN3_GiamDoc
    WHERE MAKH = p_makh AND ISDELETE = 0;
EXCEPTION WHEN OTHERS THEN
    v_count_cn3 := 0;
    v_sum_cn3 := 0;
END;

-- Tổng
v_total_hd := v_count_cn1 + v_count_cn2 + v_count_cn3;
v_total_tien := v_sum_cn1 + v_sum_cn2 + v_sum_cn3;

-- In kết quả
DBMS_OUTPUT.PUT_LINE('--- THỐNG KÊ CHO KHÁCH HÀNG: ' || p_makh || ' ---');
DBMS_OUTPUT.PUT_LINE('CN1: ' || v_count_cn1 || ' hóa đơn - ' ||
TO_CHAR(v_sum_cn1, 'FM999G999G999G999G999') || ' VNĐ');
DBMS_OUTPUT.PUT_LINE('CN2: ' || v_count_cn2 || ' hóa đơn - ' ||
TO_CHAR(v_sum_cn2, 'FM999G999G999') || ' VNĐ');
DBMS_OUTPUT.PUT_LINE('CN3: ' || v_count_cn3 || ' hóa đơn - ' ||
TO_CHAR(v_sum_cn3, 'FM999G999G999') || ' VNĐ');
DBMS_OUTPUT.PUT_LINE('=> TỔNG: ' || v_total_hd || ' hóa đơn - ' ||
TO_CHAR(v_total_tien, 'FM999G999G999G999G999') || ' VNĐ');

RETURN v_total_tien;
END;
/

SET SERVEROUTPUT ON;
DECLARE
    tong_tien NUMBER;
BEGIN
    tong_tien := ThongKeMuaHangKhachHang('KH0001');

```

```

    DBMS_OUTPUT.PUT_LINE('Tổng tiền trả về từ function: ' || TO_CHAR(tong_tien,
'FM999G999G999G999G999') || ' VNĐ');
END;
/

```

2.2. Procedure

2.2.1. Procedure Thêm 1 khách hàng mới bất kì vào 1 chi nhánh bất kì sẽ đồng bộ với các chi nhánh còn lại

```

CREATE OR REPLACE PROCEDURE DongBoKhachHang(
    p_makh      IN VARCHAR2,
    p_hoten     IN VARCHAR2,
    p_diachi    IN VARCHAR2,
    p_sdt       IN VARCHAR2,
    p_tongtien  IN NUMBER DEFAULT 0
)
IS
    v_exists NUMBER := 0;
    v_da_ton_tai BOOLEAN := FALSE;
BEGIN
    DBMS_OUTPUT.PUT_LINE('=== BẮT ĐẦU ĐỒNG BỘ KHÁCH HÀNG: ' || p_makh || '
===');

    -- CN2 (local)
    BEGIN
        SELECT COUNT(*) INTO v_exists FROM KHACHHANG WHERE MAKH = p_makh;
        IF v_exists = 0 THEN
            INSERT INTO KHACHHANG(MAKH, HOTEN, DIACHI, SODIENTHOAI, TONGTIEN,
ISDELETE)
            VALUES (p_makh, p_hoten, p_diachi, p_sdt, p_tongtien, 0);
            DBMS_OUTPUT.PUT_LINE('Đã thêm vào chi nhánh hiện tại (CN2)');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Khách hàng đã tồn tại tại CN2');
            v_da_ton_tai := TRUE;
        END IF;
    EXCEPTION WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Lỗi khi xử lý tại CN2');
    END;

    -- CN1
    BEGIN
        SELECT COUNT(*) INTO v_exists FROM CN1.KHACHHANG@CN1_GiamDoc WHERE MAKH =
p_makh;
        IF v_exists = 0 THEN
            INSERT INTO CN1.KHACHHANG@CN1_GiamDoc(MAKH, HOTEN, DIACHI, SODIENTHOAI,
TONGTIEN, ISDELETE)
            VALUES (p_makh, p_hoten, p_diachi, p_sdt, p_tongtien, 0);
            DBMS_OUTPUT.PUT_LINE('Đồng bộ vào CN1');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Khách hàng đã tồn tại tại CN1');
            v_da_ton_tai := TRUE;
        END IF;
    EXCEPTION WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Không thể truy cập hoặc đồng bộ CN1');
    END;

    -- CN3
    BEGIN

```



```

SELECT COUNT(*) INTO v_exists FROM CN3.KHACHHANG@CN3_GiamDoc WHERE MAKH =
p_makh;
IF v_exists = 0 THEN
    INSERT INTO CN3.KHACHHANG@CN3_GiamDoc(MAKH, HOTEN, DIACHI, SODIENTHOAI,
TONGTIEN, ISDELETE)
    VALUES (p_makh, p_hoten, p_diachi, p_sdt, p_tongtien, 0);
    DBMS_OUTPUT.PUT_LINE('Đồng bộ vào CN3');
ELSE
    DBMS_OUTPUT.PUT_LINE('Khách hàng đã tồn tại tại CN3');
    v_da_ton_tai := TRUE;
END IF;
EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Không thể truy cập hoặc đồng bộ CN3');
END;

IF v_da_ton_tai THEN
    DBMS_OUTPUT.PUT_LINE('Khách hàng đã có ở ít nhất một chi nhánh. Không ghi
đề. ');
END IF;

COMMIT;
END;
/

BEGIN
    DongBokhachHang(
        p_makh      => 'KH1004',
        p_hoten      => 'Nguyễn Thanh Bảo',
        p_diachi     => 'HCM',
        p_sdt        => '0909009090',
        p_tongtien   => 0
    );
END;
/

```

2.3. Ràng buộc toàn vẹn

Trigger: Kiểm tra số lượng sản phẩm tồn kho trước khi bán.

Bối cảnh: CTHD, HOADON, NHANVIEN_PUBLIC, TONKHO

Nội dung: $\forall c \in \text{CTHD}, \exists h \in \text{HOADON}, n \in \text{NHANVIEN_PUBLIC}, t \in \text{TONKHO}$:

$c.\text{MAHD} = h.\text{MAHD} \wedge h.\text{MANV} = n.\text{MANV} \wedge c.\text{MASP} = t.\text{MASP} \wedge n.\text{MACN} = t.\text{MACN}$
 $\Rightarrow c.\text{SOLUONG} \leq t.\text{SOLUONG}$

2.3.1. Bảng tầm ảnh hưởng

Bảng	Thêm	Xóa	Sửa
CTHD	+	-	+(SOLUONG)
HOADON	-	-	-
NHANVIEN_PUBLIC	-	-	-

TONKHO	-	-	-
--------	---	---	---

```

CREATE OR REPLACE TRIGGER check_inventory_before_selling
BEFORE INSERT ON CTHD
FOR EACH ROW
DECLARE
    v_soluong_tonkho TONKHO.SOLUONG%TYPE;
    v_machinhanh NHANVIEN_PUBLIC.MACHINHANH%TYPE;
BEGIN
    SELECT MACHINHANH INTO v_machinhanh
    FROM HOADON H JOIN NHANVIEN_PUBLIC NV ON H.MANV = NV.MANV
    WHERE H.MAHD = :NEW.MAHD;
    SELECT SOLUONG INTO v_soluong_tonkho
    FROM TONKHO
    WHERE MASP = :NEW.MASP AND MACHINHANH = v_machinhanh;
    IF v_soluong_tonkho < :NEW.SOLUONG THEN
        RAISE_APPLICATION_ERROR(-20001, 'Số lượng tồn kho không đủ để bán.');
```

```

    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20002, 'Không tìm thấy sản phẩm trong tồn kho tại
chi nhánh.');
```

END;

Trigger : Cập nhật số lượng sản phẩm tồn kho sau khi bán

Bối cảnh: CTHD, HOADON, NHANVIEN_PUBLIC, TONKHO

Nội dung : $\forall c \in \text{CTHD}, \exists h \in \text{HOADON}, n \in \text{NHANVIEN_PUBLIC}, t \in \text{TONKHO}$:

$c.MAHD=h.MAHD \wedge h.MANV=n.MANV \wedge c.MASP=t.MASP \wedge n.MACN=t.MACN$

$\Rightarrow t.SOLUONG := t.SOLUONG - C.SOLUONG$, $t.TINHTRANG = \text{'Hết hàng'}$ if $t.SOLUONG \leq 0$ else $t.TINHTRANG = \text{'Còn hàng'}$

Bảng tầm ảnh hưởng:

Bảng	Thêm	Xóa	Sửa
CTHD	+	-	-
HOADON	-	-	-
NHANVIEN_PUBLIC	-	-	-
TONKHO	-	-	+(SOLUONG, TINHTRANG,

			NGAYCAPNHAT)
--	--	--	--------------

```

CREATE OR REPLACE TRIGGER update_inventory_after_selling
AFTER INSERT ON CTHD
FOR EACH ROW
DECLARE
    v_machinhanh NHANVIEN_PUBLIC.MACHINHANH%TYPE;
BEGIN
    SELECT NV.MACHINHANH INTO v_machinhanh
    FROM HOADON H
    JOIN NHANVIEN_PUBLIC NV ON H.MANV = NV.MANV
    WHERE H.MAHD = :NEW.MAHD;
    UPDATE TONKHO
    SET SOLUONG = SOLUONG - :NEW.SOLUONG,
        TINHTRANG = CASE
            WHEN SOLUONG - :NEW.SOLUONG <= 0 THEN 'Hết Hàng'
            ELSE 'Còn Hàng'
        END,
        NGAYCAPNHAT = SYSDATE
    WHERE MASP = :NEW.MASP AND MACHINHANH = v_machinhanh;
END;

```

3. CÁC MỨC CÔ LẬP (ISOLATION LEVEL) TRONG MÔI TRƯỜNG PHÂN TÁN

3.1. Lost update

3.1.1. Mô tả tình huống

Tại chi nhánh **CN3**, quản lý (Session 1) đang xử lý hóa đơn bán sản phẩm SP0001 cho khách nên hệ thống giảm 1 sản phẩm khỏi kho. Cùng lúc đó, giám đốc **CN1** (Session 2) phát hiện có 2 sản phẩm SP0001 trong kho bị hư hại do ẩm mốc → cập nhật để loại bỏ 2 sản phẩm này khỏi tồn kho.

Cả hai thao tác đều dựa vào cùng một giá trị tồn kho ban đầu (70), nhưng cập nhật riêng biệt mà không biết đến nhau.

Kết quả: chỉ một cập nhật được giữ, cập nhật còn lại bị mất (overwritten) → Lost Update xảy ra.

3.1.2. Minh họa

Bảng 5. Minh họa tình huống Lost update

Thời điểm	Session 1 (Quản lý)	Session 2 (Giám đốc)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED; COMMIT;	Cả hai giao dịch dùng mức cô lập mặc định – không khóa dòng
t1	SELECT * FROM CN3.TONKHO WHERE MASP = 'SP0001'; ----- SOLUONG 70		Xem số lượng sản phẩm SP0001 trong kho CN2
t2	UPDATE CN3.TONKHO SET SOLUONG = 69 WHERE MASP = 'SP0001';	No action	Session 1 giảm 1 sản phẩm do bán cho khách
t3	(Chưa commit)	UPDATE CN3.TONKHO@CN3_GiamDoc SET SOLUONG = 68 WHERE MASP = 'SP0001';	Session 2 cập nhật giảm 2 sản phẩm do hư hỏng – không biết rằng Session 1 đã cập nhật trước đó
t4	COMMIT;	(Chưa commit)	Session 1 hoàn tất giao dịch và ghi dữ liệu (SOLUONG = 69)
t5	(Đã commit)	COMMIT;	Session 2 commit → ghi đè SoLuong thành 68
t6	SELECT SOLUONG FROM CN3.TONKHO WHERE MASP = 'SP0001'; ----- SOLUONG 68		Kết quả: 68 ← mất cập nhật giảm 1 của Session 1

3.1.3. Giải pháp

Giải pháp: Dùng SERIALIZABLE

Bảng 6. Giải pháp cho tình huống Lost update

Thời điểm	Session 1 (Quản lý)	Session 2 (Giám đốc)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	Thiết lập mức cách ly cao nhất để ngăn lỗi ghi đề
t1	SELECT * FROM CN3.TONKHO WHERE MASP = 'SP0001'; ----- SOLUONG 70		Xem số lượng sản phẩm SP0001 trong kho CN2
t1	UPDATE CN3.TONKHO SET SOLUONG = 69 WHERE MASP = 'SP0001';	No action	Session 1 giảm 1 sản phẩm
t2	(Chưa commit)	UPDATE CN3.TONKHO@CN3_GiamDoc SET SOLUONG = 68 WHERE MASP = 'SP0001';	Oracle kiểm tra và phát hiện xung đột , vì dòng này đã bị thay đổi trong phiên chưa commit
t3	COMMIT;	ORA-08177: can't serialize access for this transaction	Session 2 bị lỗi → không thể ghi đề dữ liệu, đảm bảo an toàn
t4		COMMIT;	
t5	SELECT SOLUONG FROM CN3.TONKHO WHERE MASP = 'SP0001'; ----- SOLUONG 69		Kết quả: 69 ← giữ đúng cập nhật của Session 1, không mất dữ liệu

3.2. Phantom Read

3.2.1. Mô tả tình huống

Quản lý tại chi nhánh CN3 muốn thống kê **số lượng hóa đơn trong tháng 5** để tổng hợp báo cáo doanh thu. Quản lý thực hiện truy vấn và thấy có **27304** hóa đơn. Trong lúc đó, một nhân viên khác tại CN3 đang tạo hóa đơn mới cho khách, hóa đơn được ghi nhận

cũng thuộc tháng 5 và được commit. Quản lý (trong cùng phiên giao dịch) **thực hiện lại truy vấn y chang trước đó** để xuất file báo cáo → lần này thấy **27305** hóa đơn.

→ **Phantom Read** xảy ra.

3.2.2. Minh họa

Bảng 7. Minh họa cho tình huống Phantom Read

Thời điểm	Session 1 (Quản lý)	Session 2 (Giám đốc)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	Cả hai session dùng mức cô lập mặc định
t1	SELECT COUNT(H.MAHD) AS SOHOADON FROM CN3.HOADON H WHERE MONTH(NGAYTAO) = 5; ----- 27304	No action	Quản lý ghi nhận có x hóa đơn trong tháng 5
t2	No action	INSERT INTO CN3.HOADON@CN3_GiamDoc (MAHD, MANV, MAKH, NGAYTAO, TONGTIEN, ISDELETE) VALUES ('HD00000', 'NV0721', 'KH0987', TO_DATE('2025/05/27', 'yyyy/mm/dd'), 0, 0);	Giám đốc tạo hóa đơn mới
t3	No action	COMMIT;	Hóa đơn mới được ghi nhận – cũng thuộc tháng 5
t4	SELECT COUNT(H.MAHD) FROM CN3.HOADON H WHERE MONTH(NGAYTAO) = 5; ----- 27305	No action	Quản lý truy vấn lại y chang → kết quả khác với lần đầu dù cùng một transaction

3.2.3. Giải pháp

Sử dụng Serializable

Bảng 8. Giải pháp cho tình huống Phantom Read

Thời điểm	Session 1 (Quản lý)	Session 2 (Nhân viên)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	Cả hai dùng mức cô lập cao nhất
t1	SELECT COUNT(H.MAHD) FROM CN3.HOADON H WHERE MONTH(NGAYTAO) = 5; ----- 27304	No action	Quản lý thống kê số hóa đơn tháng 5
t2		INSERT INTO CN3.HOADON@CN3_GiamDoc (MAHD, MANV, MAKH, NGAYTAO, TONGTIEN, ISDELETE) VALUES ('HD00000', 'NV0721', 'KH0987', TO_DATE('2025/05/27', 'yyyy/mm/dd'), 0, 0);	Giám đốc cố gắng thêm hóa đơn mới
t3			Oracle từ chối do phạm vi dữ liệu đã bị đọc – ngăn chặn thay đổi
t4	SELECT COUNT(H.MAHD) FROM CN3.HOADON H WHERE MONTH(NGAYTAO) = 5; ----- 27304		Không xuất hiện dữ liệu "bóng ma"

3.3. Dirty Read

Trong hệ quản trị cơ sở dữ liệu Oracle, mức cô lập mặc định là read committed vì vậy không bao giờ có trường hợp Dirty Read.

3.4. Unrepeatable Read

3.4.1. Mô tả tình huống

- Nhân viên tại **CN2 (Session 1)** mở giao diện sản phẩm để tư vấn cho khách hàng cần mua số lượng lớn SP0002. Hệ thống hiển thị rằng sản phẩm SP0002 tại **CN2** đã hết mà còn **12 cái** trong kho của **CN1**. Nên nhân viên này đã giới thiệu cho khách sang **CN1**.

- Cùng lúc đó, giám đốc tại **CN1 (Session 2)** cập nhật tồn kho còn **1** và COMMIT.
- Khi kiểm tra lại, hệ thống báo sản phẩm SP0002 tại kho **CN1** còn 1 sản phẩm.

3.4.2. Minh họa

Bảng 9. Minh họa tình huống Unrepeatable Read

Thời điểm	Session 1 (Nhân viên tại CN2)	Session 2 (Giám đốc)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	Cả hai session đều sử dụng mức cô lập mặc định là READ COMMITTED, cho phép đọc dữ liệu đã commit của các giao dịch khác.
t1	SELECT * FROM CN1.TONKHO@CN1_NhanVien2 WHERE MASP = 'SP0002'; ----- SOLUONG 12	No action	Nhân viên tại CN2 (Session 1) truy vấn tồn kho sản phẩm SP0002 tại CN1 → ghi nhận còn 12 sản phẩm.
t2	No action	UPDATE CN1.TONKHO SET SOLUONG = 1 WHERE MASP = 'SP0002';	Giám đốc tại CN1 (Session 2) cập nhật số lượng SP0002 xuống 1 và COMMIT.
t3		COMMIT;	
t4	SELECT * FROM CN1.TONKHO@CN1_NhanVien2 WHERE MASP = 'SP0002'; ----- SOLUONG 1	No action	Session 1 đọc lại dữ liệu tồn kho SP0002 tại CN1 → kết quả là 1, khác với kết quả ban đầu (2) → xảy ra lỗi Unrepeatable Read.

3.4.3. Giải pháp

Bảng 10. Giải pháp cho tình huống Unrepeatable Read

Thời điểm	Session 1 (Nhân viên tại CN2)	Session 2 (Giám đốc)	Giải thích
t0	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	ALTER SESSION SET ISOLATION_LEVEL = SERIALIZABLE;	Cả hai session đều đặt mức ISOLATION_LEVEL = SERIALIZABLE để đảm

			bảo dữ liệu không thay đổi trong suốt transaction.
t1	<pre>SELECT * FROM CN1.TONKHO@CN1_NhanVien2 WHERE MASP = 'SP0002'; ----- SOLUONG 12</pre>		Session 1 truy vấn tồn kho SP0002 tại CN1 và ghi nhận còn 12 sản phẩm.
t2		<pre>UPDATE CN1.TONKHO SET SOLUONG = 1 WHERE MASP = 'SP0002';</pre>	Session 2 cố gắng cập nhật số lượng sản phẩm SP0002 về 1 tại CN1
t3		ORA-08177: can't serialize access for this transaction	Oracle phát hiện dữ liệu đã bị đọc bởi một transaction khác chưa kết thúc → từ chối ghi và trả về lỗi ORA-08177.
t4	<pre>SELECT * FROM CN1.TONKHO@CN1_NhanVien2 WHERE MASP = 'SP0002'; ----- SOLUONG 12</pre>		Session 1 đọc lại dữ liệu SP0002 vẫn thấy 12 sản phẩm như ban đầu → không xảy ra Unrepeatable Read.
T5	COMMIT;		Session 1 kết thúc giao dịch bằng COMMIT; sau đó các session khác mới có thể cập nhật dòng đó.

3.5. Deadlock

3.5.1. Mô tả tình huống

- **Quản lý CN3 (QuanLy3)** cập nhật sản phẩm SP0001, sau đó chuyển sang SP0002.
- **Giám đốc (GiamDoc)** cũng cập nhật, nhưng ngược thứ tự: SP0002 trước rồi SP0001.

3.5.2. Minh họa

Thời điểm	Session 1 (Quản lý tại CN3)	Session 2 (Giám đốc)	Giải thích
-----------	-----------------------------	----------------------	------------

t0	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	ALTER SESSION SET ISOLATION_LEVEL = READ COMMITTED;	Dùng mức mặc định
t1	UPDATE CN3.SANPHAM SET GIABAN = 2000000 WHERE MASP = 'SP0001';	No action	QuanLy3 giữ khóa SP001
t2	(Giữ SP0001)	UPDATE CN3.SANPHAM@CN3_GiamDoc SET GIABAN = 1500000 WHERE MASP = 'SP0002';	GiamDoc giữ khóa SP0002
t3	UPDATE CN3.SANPHAM SET GIABAN = 2200000 WHERE MASP = 'SP0002';	(Giữ SP0002)	QuanLy3 chờ SP0002 → bị chặn
t4	(Chờ SP0002)	UPDATE CN3.SANPHAM@CN3_GiamDoc SET GIABAN = 2100000 WHERE MASP = 'SP0001';	GiamDoc chờ SP0001 → bị chặn
t5			DEADLOCK chuẩn do khóa chéo SP0001 ↔ SP0002

3.5.3. Giải pháp

Để giải quyết deadlock, chỉ cần thực hiện commit.

4. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

4.1. Câu truy vấn SQL chưa tối ưu

Liệt kê thông tin hóa đơn (mã hóa đơn, tên sản phẩm, tên khách hàng, địa chỉ khách hàng, tổng tiền, ngày tạo, số lượng) của CN1 trong năm 2024 có tổng tiền trên 5 triệu và bán sản phẩm thuộc thương hiệu Yonex.

```
SELECT *
FROM (
  -- CN1
  SELECT
    H.MAHD AS MAHD,
    SP.TENSANPHAM AS TENSANPHAM,
    C.SOLUONG AS SOLUONG,
    KH.HOTEN AS TEN_KHACH,
    KH.DIACHI AS DIACHI_KH,
    H.TONGTIEN AS TONGTIEN,
    H.NGAYTAO AS NGAYTAO,
```

```

NV.MACHINHANH AS MACHINHANH,
H.ISDELETE AS HD_DEL,
C.ISDELETE AS CTHD_DEL,
SP.ISDELETE AS SP_DEL,
KH.ISDELETE AS KH_DEL,
NV.ISDELETE AS NV_DEL,
SP.THUONGHIEU AS THUONGHIEU
FROM CN1.HOADON H, CN1.CTHD C, CN1.SANPHAM SP, CN1.KHACHHANG KH,
CN1.NHANVIEN_PUBLIC NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV

UNION ALL

-- CN2
SELECT
H.MAHD, SP.TENSANPHAM, C.SOLUONG,
KH.HOTEN, KH.DIACHI, H.TONGTIEN, H.NGAYTAO, NV.MACHINHANH,
H.ISDELETE, C.ISDELETE, SP.ISDELETE, KH.ISDELETE, NV.ISDELETE,
SP.THUONGHIEU
FROM CN2.HOADON@CN2_GiamDoc H, CN2.CTHD@CN2_GiamDoc C,
CN2.SANPHAM@CN2_GiamDoc SP, CN2.KHACHHANG@CN2_GiamDoc KH,
CN2.NHANVIEN_PUBLIC@CN2_GiamDoc NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV

UNION ALL

-- CN3
SELECT
H.MAHD, SP.TENSANPHAM, C.SOLUONG,
KH.HOTEN, KH.DIACHI, H.TONGTIEN, H.NGAYTAO, NV.MACHINHANH,
H.ISDELETE, C.ISDELETE, SP.ISDELETE, KH.ISDELETE, NV.ISDELETE,
SP.THUONGHIEU
FROM CN3.HOADON@CN3_GiamDoc H, CN3.CTHD@CN3_GiamDoc C,
CN3.SANPHAM@CN3_GiamDoc SP, CN3.KHACHHANG@CN3_GiamDoc KH,
CN3.NHANVIEN_PUBLIC@CN3_GiamDoc NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV
)
WHERE HD_DEL = 0 AND CTHD_DEL = 0 AND SP_DEL = 0
AND KH_DEL = 0 AND NV_DEL = 0
AND TONGTIEN > 5000000
AND EXTRACT(YEAR FROM NGAYTAO) = 2024
AND THUONGHIEU = 'Yonex'
AND MACHINHANH = 'CN1';

```

4.2. Thực hiện EXPLAIN câu truy vấn

```
-- EXPLAIN kế hoạch thực thi
EXPLAIN PLAN FOR
SELECT *
FROM (
  -- CN1
  SELECT
    H.MAHD AS MAHD,
    SP.TENSANPHAM AS TENSANPHAM,
    C.SOLUONG AS SOLUONG,
    KH.HOTEN AS TEN_KHACH,
    KH.DIACHI AS DIACHI_KH,
    H.TONGTIEN AS TONGTIEN,
    H.NGAYTAO AS NGAYTAO,
    NV.MACHINHANH AS MACHINHANH,
    H.ISDELETE AS HD_DEL,
    C.ISDELETE AS CTHD_DEL,
    SP.ISDELETE AS SP_DEL,
    KH.ISDELETE AS KH_DEL,
    NV.ISDELETE AS NV_DEL,
    SP.THUONGHIEU AS THUONGHIEU
  FROM CN1.HOADON H, CN1.CTHD C, CN1.SANPHAM SP, CN1.KHACHHANG KH,
  CN1.NHANVIEN_PUBLIC NV
  WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
    AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV

  UNION ALL

  -- CN2
  SELECT
    H.MAHD, SP.TENSANPHAM, C.SOLUONG,
    KH.HOTEN, KH.DIACHI, H.TONGTIEN, H.NGAYTAO, NV.MACHINHANH,
    H.ISDELETE, C.ISDELETE, SP.ISDELETE, KH.ISDELETE, NV.ISDELETE,
    SP.THUONGHIEU
  FROM CN2.HOADON@CN2_GiamDoc H, CN2.CTHD@CN2_GiamDoc C,
    CN2.SANPHAM@CN2_GiamDoc SP, CN2.KHACHHANG@CN2_GiamDoc KH,
    CN2.NHANVIEN_PUBLIC@CN2_GiamDoc NV
  WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
    AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV

  UNION ALL

  -- CN3
  SELECT
    H.MAHD, SP.TENSANPHAM, C.SOLUONG,
    KH.HOTEN, KH.DIACHI, H.TONGTIEN, H.NGAYTAO, NV.MACHINHANH,
    H.ISDELETE, C.ISDELETE, SP.ISDELETE, KH.ISDELETE, NV.ISDELETE,
    SP.THUONGHIEU
  FROM CN3.HOADON@CN3_GiamDoc H, CN3.CTHD@CN3_GiamDoc C,
```



```

CN3.SANPHAM@CN3_GiamDoc SP, CN3.KHACHHANG@CN3_GiamDoc KH,
CN3.NHANVIEN_PUBLIC@CN3_GiamDoc NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP
AND H.MAKH = KH.MAKH AND H.MANV = NV.MANV
)
WHERE HD_DEL = 0 AND CTHD_DEL = 0 AND SP_DEL = 0
AND KH_DEL = 0 AND NV_DEL = 0
AND TONGTIEN > 5000000
AND EXTRACT(YEAR FROM NGAYTAO) = 2024
AND THUONGHIEU = 'Yonex'
AND MACHINHANH = 'CN1';

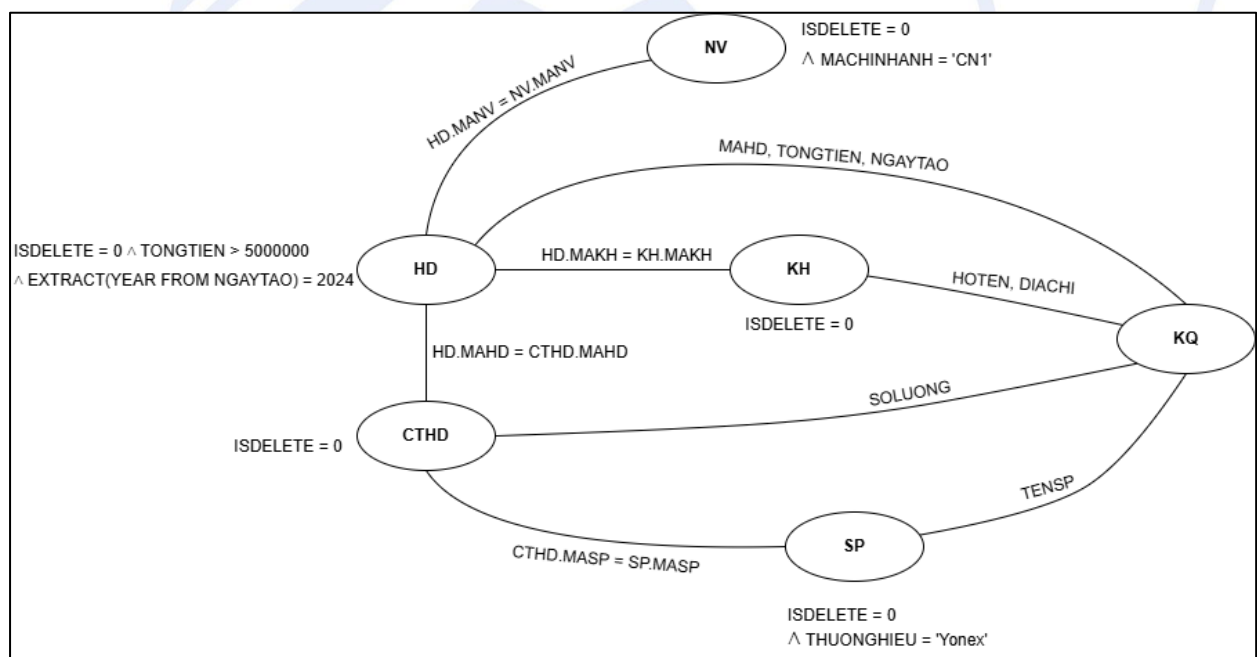
-- Xem kế hoạch thực thi
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

```

4.3. Tối ưu hóa câu truy vấn cục bộ, phân tán

4.3.1. Tối ưu hoá câu truy vấn cục bộ

4.3.1.1. Đồ thị truy vấn



Hình 4-1. Đồ thị truy vấn

⇒ Đồ thị truy vấn liên thông nên câu truy vấn đúng ngữ nghĩa.

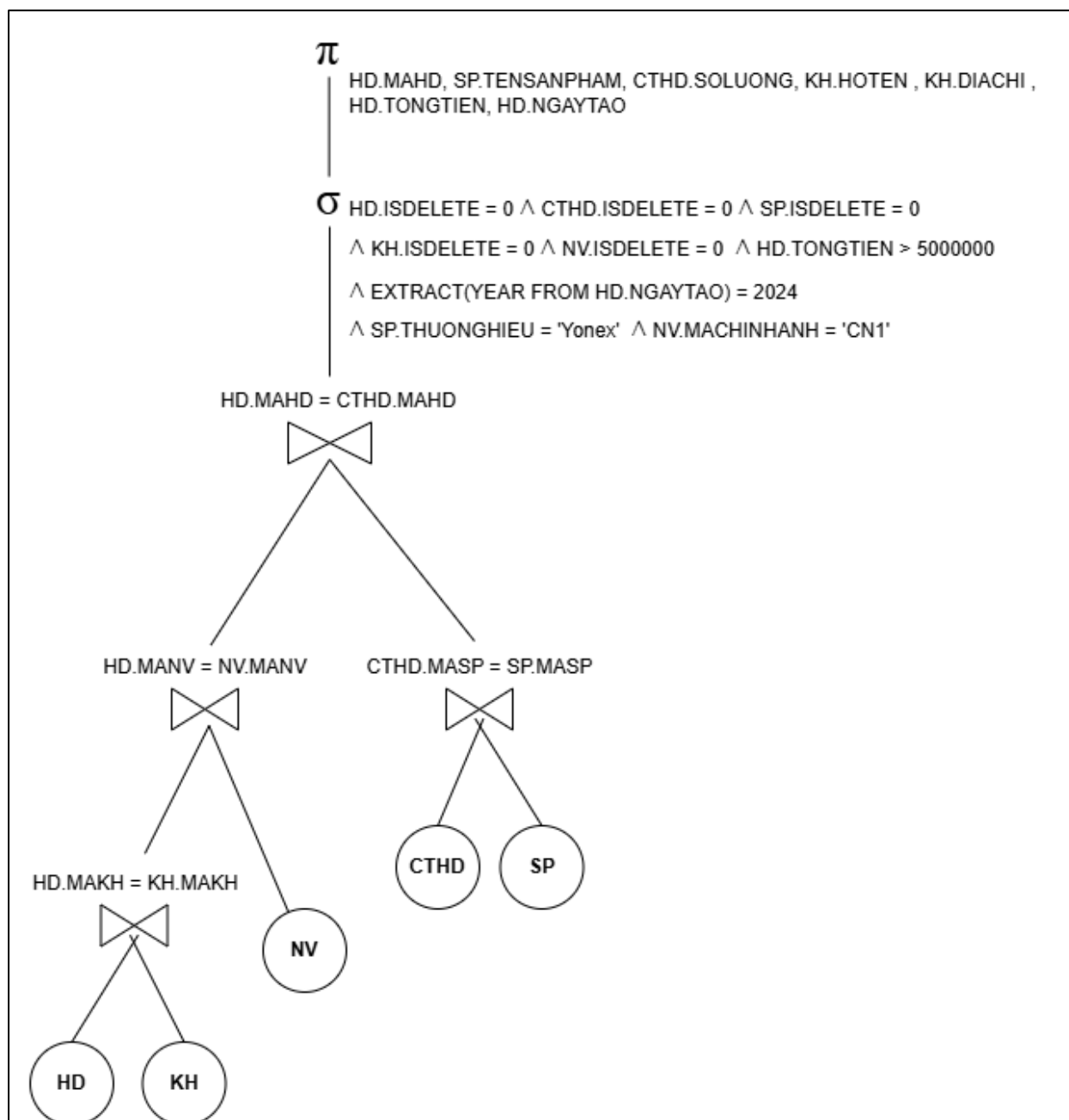
Bảng 11. Bảng viết tắt

Từ viết tắt	Từ đầy đủ
HD	Quan hệ HOADON
NV	Quan hệ NHANVIEN_PUBLIC
KH	Quan hệ KHACHHANG

SP	Quan hệ SANPHAM
KQ	Kết quả

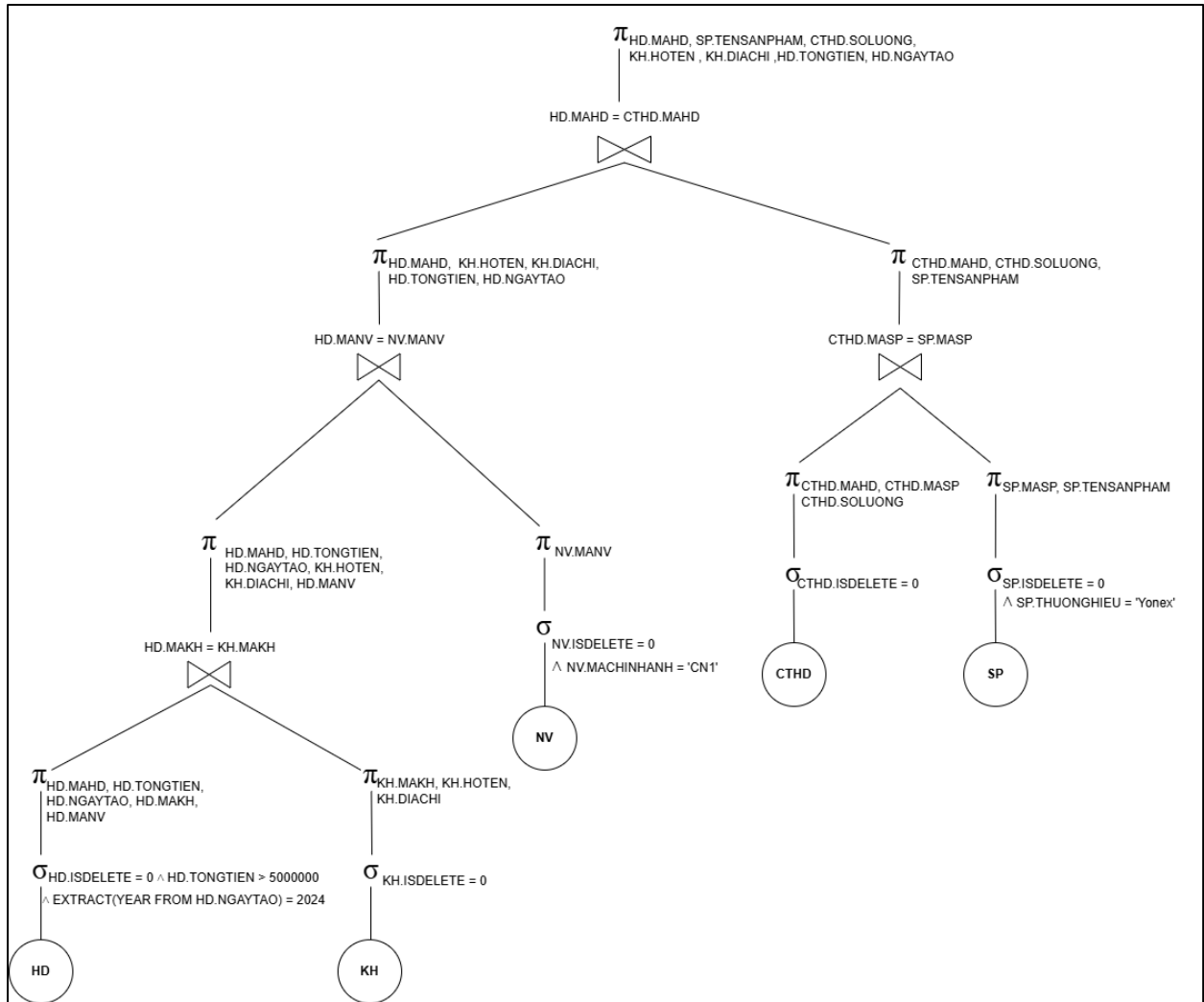
4.3.1.2. Tối ưu hoá câu truy vấn cục bộ

Câu truy vấn ban đầu:



Hình 4-2. Cây truy vấn ban đầu

Câu truy vấn sau khi tối ưu toàn cục:



Hình 4-3. Cây truy vấn tối ưu toàn cục

4.3.1.3. Câu truy vấn đã tối ưu

```

SELECT *
FROM (
  -- CN1
  SELECT H.MAHD, SP.TENSANPHAM, C.SOLUONG, KH.HOTEN, KH.DIACHI, H.TONGTIEN,
  H.NGAYTAO
  FROM
    (SELECT MAHD, TONGTIEN, NGAYTAO, MAKH, MANV
     FROM CN1.HOADON
     WHERE ISDELETE = 0 AND TONGTIEN > 5000000 AND EXTRACT(YEAR FROM NGAYTAO)
     = 2024) H,

    (SELECT MAKH, HOTEN, DIACHI
     FROM CN1.KHACHHANG
     WHERE ISDELETE = 0) KH,

    (SELECT MAHD, MASP, SOLUONG
     FROM CN1.CTHD
     WHERE ISDELETE = 0) C,

    (SELECT MASP, TENSANPHAM

```

```

FROM CN1.SANPHAM
WHERE ISDELETE = 0 AND THUONGHIEU = 'Yonex') SP,

(SELECT MANV
FROM CN1.NHANVIEN_PUBLIC
WHERE ISDELETE = 0 AND MACHINHANH = 'CN1') NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP AND H.MAKH = KH.MAKH AND H.MANV
= NV.MANV

UNION ALL

-- CN2
SELECT H.MAHD, SP.TENSANPHAM, C.SOLUONG, KH.HOTEN, KH.DIACHI, H.TONGTIEN,
H.NGAYTAO
FROM
(SELECT MAHD, TONGTIEN, NGAYTAO, MAKH, MANV
FROM CN2.HOADON@CN2_GiamDoc
WHERE ISDELETE = 0 AND TONGTIEN > 5000000 AND EXTRACT(YEAR FROM NGAYTAO)
= 2024) H,

(SELECT MAKH, HOTEN, DIACHI
FROM CN2.KHACHHANG@CN2_GiamDoc
WHERE ISDELETE = 0) KH,

(SELECT MAHD, MASP, SOLUONG
FROM CN2.CTHD@CN2_GiamDoc
WHERE ISDELETE = 0) C,

(SELECT MASP, TENSANPHAM
FROM CN2.SANPHAM@CN2_GiamDoc
WHERE ISDELETE = 0 AND THUONGHIEU = 'Yonex') SP,

(SELECT MANV
FROM CN2.NHANVIEN_PUBLIC@CN2_GiamDoc
WHERE ISDELETE = 0 AND MACHINHANH = 'CN1') NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP AND H.MAKH = KH.MAKH AND H.MANV
= NV.MANV

UNION ALL

-- CN3
SELECT H.MAHD, SP.TENSANPHAM, C.SOLUONG, KH.HOTEN, KH.DIACHI, H.TONGTIEN,
H.NGAYTAO
FROM
(SELECT MAHD, TONGTIEN, NGAYTAO, MAKH, MANV
FROM CN3.HOADON@CN3_GiamDoc
WHERE ISDELETE = 0 AND TONGTIEN > 5000000 AND EXTRACT(YEAR FROM NGAYTAO)
= 2024) H,

(SELECT MAKH, HOTEN, DIACHI

```

```
FROM CN3.KHACHHANG@CN3_GiamDoc
WHERE ISDELETE = 0) KH,

(SELECT MAHD, MASP, SOLUONG
FROM CN3.CTHD@CN3_GiamDoc
WHERE ISDELETE = 0) C,

(SELECT MASP, TENSANPHAM
FROM CN3.SANPHAM@CN3_GiamDoc
WHERE ISDELETE = 0 AND THUONGHIEU = 'Yonex') SP,

(SELECT MANV
FROM CN3.NHANVIEN_PUBLIC@CN3_GiamDoc
WHERE ISDELETE = 0 AND MACHINHANH = 'CN1') NV
WHERE H.MAHD = C.MAHD AND C.MASP = SP.MASP AND H.MAKH = KH.MAKH AND H.MANV
= NV.MANV
);WHERE H.MAHD = C.MAHD
AND C.MASP = SP.MASP
AND H.MAKH = KH.MAKH
AND H.MANV = NV.MANV;
```

4.3.1.4. Thời gian truy vấn khi chưa tối ưu

PLAN_TABLE_OUTPUT VARCHAR2(4000)								

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT

0	SELECT STATEMENT		3260	1263K	1597 (3)	00:00:01		
1	VIEW		3260	1263K	1597 (3)	00:00:01		
2	UNION-ALL							
* 3	HASH JOIN		3260	557K	1597 (3)	00:00:01		
* 4	TABLE ACCESS FULL	SANPHAM	579	37635	7 (0)	00:00:01		
* 5	HASH JOIN		7069	759K	1589 (3)	00:00:01		
* 6	HASH JOIN		2617	227K	574 (4)	00:00:01		
* 7	TABLE ACCESS FULL	KHACHHANG	1000	37000	5 (0)	00:00:01		
* 8	HASH JOIN		2618	132K	569 (4)	00:00:01		
* 9	TABLE ACCESS FULL	NHANVIEN_PUBLIC	333	4329	5 (0)	00:00:01		
* 10	TABLE ACCESS FULL	HOADON	2626	100K	564 (4)	00:00:01		
* 11	TABLE ACCESS FULL	CTHD	899K	18M	1009 (2)	00:00:01		
12	REMOTE					CN2_G~	R->S	
13	REMOTE					CN3_G~	R->S	

Hình 4-4. Thời gian truy vấn khi chưa tối ưu

4.3.1.5. Thời gian truy vấn khi đã tối ưu

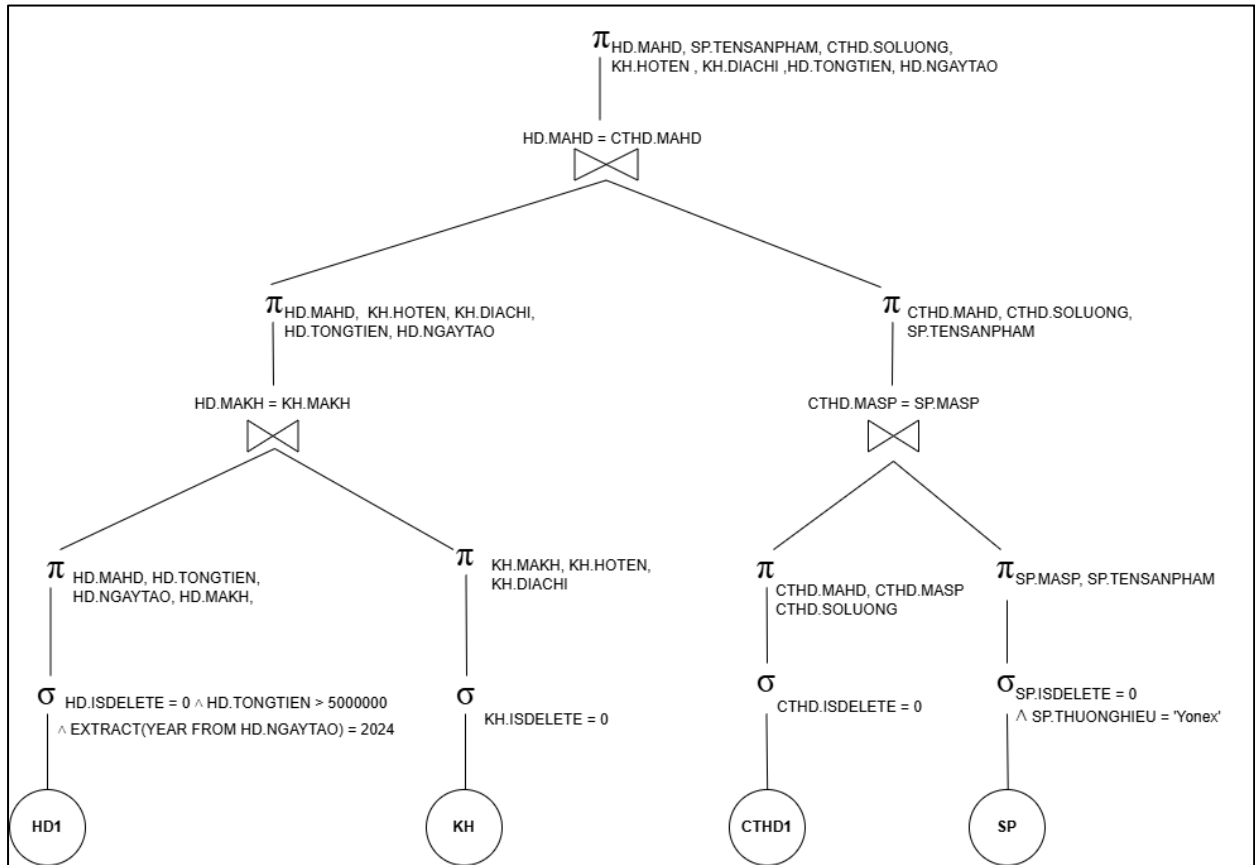
PLAN_TABLE_OUTPUT VARCHAR2(4000)									
Plan hash value: 1090798549									

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT	

0	SELECT STATEMENT		3260	948K	1597 (3)	00:00:01			
1	VIEW		3260	948K	1597 (3)	00:00:01			
2	UNION-ALL								
* 3	HASH JOIN		3260	557K	1597 (3)	00:00:01			
* 4	TABLE ACCESS FULL	SANPHAM		579	37635	7 (0)	00:00:01		
* 5	HASH JOIN		7069	759K	1589 (3)	00:00:01			
* 6	HASH JOIN		2617	227K	574 (4)	00:00:01			
* 7	TABLE ACCESS FULL	KHACHHANG		1000	37000	5 (0)	00:00:01		
* 8	HASH JOIN		2618	132K	569 (4)	00:00:01			
* 9	TABLE ACCESS FULL	NHANVIEN_PUBLIC		333	4329	5 (0)	00:00:01		
* 10	TABLE ACCESS FULL	HOADON		2626	100K	564 (4)	00:00:01		
* 11	TABLE ACCESS FULL	CTHD		899K	18M	1009 (2)	00:00:01		
12	REMOTE					CN2_G~	R->S		
13	REMOTE					CN3_G~	R->S		

Hình 4-5. Thời gian truy vấn sau khi tối ưu

4.3.2. Cây truy vấn rút gọn trên các mảnh



Hình 4-6. Cây truy vấn rút gọn trên các mảnh

4.4. Viết lại câu truy vấn rút gọn trên các mảnh

```

SELECT H.MAHD, SP.TENSANPHAM, C.SOLUONG, KH.HOTEN AS TEN_KHACH, KH.DIACHI AS
DIACHI_KH, H.TONGTIEN, H.NGAYTAO
FROM
  (SELECT MAHD, TONGTIEN, NGAYTAO, MAKH
   FROM CN1.HOADON
   WHERE ISDELETE = 0 AND TONGTIEN > 5000000
    AND EXTRACT(YEAR FROM NGAYTAO) = 2024) H,

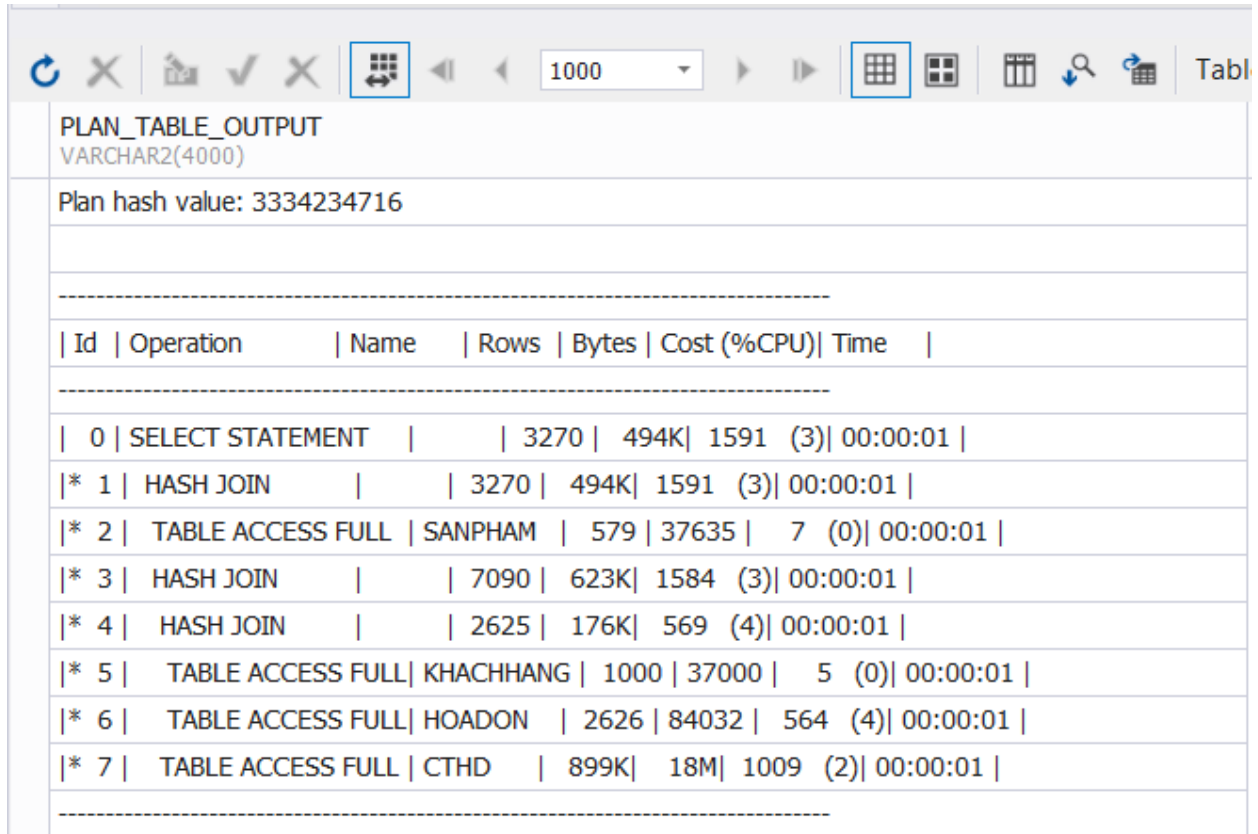
  (SELECT MAKH, HOTEN, DIACHI
   FROM CN1.KHACHHANG
   WHERE ISDELETE = 0) KH,

  (SELECT MAHD, MASP, SOLUONG
   FROM CN1.CTHD
   WHERE ISDELETE = 0) C,

  (SELECT MASP, TENSANPHAM
   FROM CN1.SANPHAM
   WHERE ISDELETE = 0 AND THUONGHIEU = 'Yonex') SP
WHERE H.MAHD = C.MAHD AND H.MAKH = KH.MAKH AND C.MASP = SP.MASP;

```

Kết quả EXPLAIN câu truy vấn sau khi rút gọn trên các mảnh:



Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		3270	494K	1591 (3)	00:00:01
* 1	HASH JOIN		3270	494K	1591 (3)	00:00:01
* 2	TABLE ACCESS FULL	SANPHAM	579	37635	7 (0)	00:00:01
* 3	HASH JOIN		7090	623K	1584 (3)	00:00:01
* 4	HASH JOIN		2625	176K	569 (4)	00:00:01
* 5	TABLE ACCESS FULL	KHACHHANG	1000	37000	5 (0)	00:00:01
* 6	TABLE ACCESS FULL	HOADON	2626	84032	564 (4)	00:00:01
* 7	TABLE ACCESS FULL	CTHD	899K	18M	1009 (2)	00:00:01

Hình 4-7. Kết quả Explain câu truy vấn.

Nhận xét:

- Chi phí thực thi tổng thể (**Cost**) giảm từ **1597** xuống còn **1591**, phản ánh rõ hiệu quả trong việc loại bỏ truy cập phân tán không cần thiết và rút gọn số lượng phép JOIN.
- Tổng lượng dữ liệu truy xuất (Bytes) giảm mạnh từ 1263K xuống 494K, cho thấy lượng dữ liệu trung gian đã được lọc bỏ sớm và hợp lý hơn.
- Ngoài ra, các phép REMOTE truy cập từ xa tới CN2 và CN3 đã được loại bỏ hoàn toàn, giúp tăng tốc độ xử lý và ổn định của hệ thống.

5. CHỈ MỤC TRONG ORACLE

Indexing là cơ chế tổ chức dữ liệu giúp tăng tốc độ truy xuất thông tin từ bảng trong cơ sở dữ liệu. Một chỉ mục (index) chứa các khóa tìm kiếm và con trỏ đến bản ghi dữ liệu thực tế. Chỉ mục giúp hệ quản trị cơ sở dữ liệu (DBMS) tránh việc quét toàn bộ bảng khi tìm kiếm.

Các loại chỉ mục:

- Chỉ mục B-tree (B-tree Index)
 - + Chỉ mục B-tree chuẩn
 - + Reverse Key Index
 - + Descending Index
 - + Key-Compressed Index
 - + Index-Organized Table (IOT)
 - + Clustered Index
- Chỉ mục Bitmap (Bitmap Index)
 - + Bitmap Join Index
- Chỉ mục dựa trên hàm (Function-Based Index)
- Chỉ mục miền (Domain Index)
 - + Oracle Text Index
 - + Spatial Index
 - + XML Index
- Chỉ mục tổng hợp (Composite Index)
- Chỉ mục duy nhất (Unique Index)
- Chỉ mục toàn cục và cục bộ (Global & Local Index)
- Chỉ mục Hash Cluster
- Các chỉ mục đặc biệt:
 - + Invisible Index
 - + Virtual Column Index

5.1. Lý thuyết về các chỉ mục

5.1.1. Chỉ mục B-tree (B-tree Index)

- **Ý nghĩa:** Là loại chỉ mục **mặc định** trong Oracle, tổ chức dữ liệu dạng cây cân bằng (B-tree) để tối ưu truy vấn..
- **Công dụng:**

- + Tìm kiếm nhanh trên cột có giá trị phân biệt cao (high cardinality) như ID, mã số.
- + Hỗ trợ truy vấn chính xác (=) và phạm vi (BETWEEN, >, <)
- **Đặc điểm:**
 - + Phù hợp cho hệ thống OLTP (xử lý giao dịch trực tuyến) do tốc độ đọc/ghi nhanh.
 - + Dữ liệu được lưu theo thứ tự tăng dần của khóa, hỗ trợ sắp xếp kết quả truy vấn.
 - + Tự động tạo khi định nghĩa PRIMARY KEY hoặc UNIQUE.
- **Ưu điểm:**
 - + Truy vấn nhanh với điều kiện WHERE, giảm thời gian quét toàn bộ bảng.
 - + Hỗ trợ truy vấn theo phạm vi (BETWEEN, <, >).
 - + Cấu trúc cân bằng giúp thời gian truy xuất ổn định dù dữ liệu lớn.
- **Nhược điểm:**
 - + Không hiệu quả với cột có ít giá trị phân biệt (low cardinality).
 - + Tăng chi phí lưu trữ và làm chậm thao tác INSERT/UPDATE/DELETE do phải cập nhật cây
- **Cấu trúc:**
 - + Root node: Node gốc, chứa con trỏ đến các branch node .
 - + Branch nodes: Lớp trung gian, chia dữ liệu thành các khoảng giá trị (ví dụ: 0-10, 11-20) .
 - + Leaf nodes: Lá chứa giá trị khóa và ROWID trỏ đến dữ liệu thực tế trong bảng .
 - + Chiều cao (height) của cây quyết định số lần I/O khi truy vấn
- **Cách sử dụng:**

```
CREATE INDEX idx_name ON table_name(column_name);
```

5.1.2. Chỉ mục Bitmap (Bitmap Index)

- **Ý nghĩa:** Chỉ mục lưu trữ thông tin dạng bitmap, mỗi giá trị cột tương ứng với một chuỗi bit.
- **Công dụng:** Tối ưu cho truy vấn phân tích trên cột có ít giá trị phân biệt (low cardinality).
- **Đặc điểm:**
 - + Thường dùng trong hệ thống OLAP hoặc Data Warehouse.
 - + Không phù hợp với bảng có nhiều thao tác ghi (INSERT/UPDATE/DELETE).
- **Ưu điểm:**
 - + Hỗ trợ truy vấn tổng hợp, phân tích dữ liệu lớn rất hiệu quả
 - + Kết hợp nhiều điều kiện OR/AND nhanh chóng.
- **Nhược điểm:**
 - + Không phù hợp cho OLTP hoặc bảng thường xuyên thay đổi dữ liệu.
- **Cấu trúc:** Mỗi giá trị cột có một bitmap, mỗi bit đại diện cho một hàng trong bảng.
- **Cách sử dụng:**

```
CREATE BITMAP INDEX idx_name ON table_name(column_name);
```

5.1.3. Index-Organized Table (IOT)

- **Ý nghĩa:** Bảng được lưu trữ dưới dạng cấu trúc B-tree của khóa chính, kết hợp cả dữ liệu và chỉ mục trong cùng một cấu trúc.
- **Công dụng:**
 - + Tối ưu truy vấn qua khóa chính với thời gian truy xuất gần như tức thì.
 - + Giảm I/O do không cần truy cập riêng index và data segment.
- **Đặc điểm:**

- + Không có bảng vật lý riêng: Dữ liệu và chỉ mục được lưu chung trong B-tree.
- + Bắt buộc có khóa chính duy nhất: Đảm bảo tính toàn vẹn dữ liệu.
- + Hỗ trợ Overflow Area: Tách dữ liệu không khóa sang segment khác nếu vượt ngưỡng PCTTHRESHOLD.
- **Ưu điểm:**
 - + Tiết kiệm không gian (không lưu trùng ROWID).
 - + Hỗ trợ tái tổ chức trực tuyến (online reorganization).
 - + Phù hợp cho OLTP và ứng dụng cần truy xuất nhanh qua khóa chính.
- **Nhược điểm:**
 - + Không hỗ trợ phân vùng (partitioning).
 - + Truy vấn qua secondary index chậm hơn do không có ROWID vật lý.
 - + Hạn chế số lượng cột không khóa do kích thước block.
- **Cấu trúc:** Cây B-tree chứa cả dữ liệu dòng.
- **Cách sử dụng:**

```
CREATE TABLE table_name (  
    column1 NUMBER,  
    column2 VARCHAR2(100),  
    PRIMARY KEY (column1)  
) ORGANIZATION INDEX;
```

5.1.4. Chỉ mục đảo ngược khoá (Reverse Key Index)

- **Ý nghĩa:** Reverse Key Index **vẫn là chỉ mục B-tree**, nhưng các byte của khóa được đảo ngược trước khi lưu vào index.
- **Công dụng:**
 - + Giảm **block contention** (tranh chấp block) khi nhiều session cùng ghi vào các khóa tăng dần (sequence-based) trong môi trường RAC hoặc OLTP.
 - + Phân tán dữ liệu đều trên các block, tránh hiện tượng "hot block" ở leaf block cuối cùng của index.
- **Đặc điểm:**

- + Không hỗ trợ Index Range Scan cho các truy vấn phạm vi (BETWEEN, >, <, LIKE 'A%').
- + Không tối ưu cho sắp xếp (ORDER BY), vì dữ liệu trong index không được lưu theo thứ tự tự nhiên.
- + Không thể sử dụng cho index-only scans nếu truy vấn yêu cầu sắp xếp.
- **Ưu điểm:**
 - + Giảm buffer busy waits và read by other session trong RAC
 - + Hiệu quả cho bảng có khóa chính tăng dần (sequence-generated PK).
- **Nhược điểm:**
 - + Tăng redo log size do cấu trúc index phức tạp hơn
 - + Không thể rebuild thành index thường mà không xóa và tạo lại.
- **Cấu trúc:** Không thay đổi giá trị cột gốc, chỉ ảnh hưởng đến cách lưu trữ trong index.
- **Cách sử dụng:**

```
CREATE INDEX idx_name ON table_name(column_name) REVERSE;
```

5.1.5. Chỉ mục dựa trên hàm (Function-Based Index)

- **Ý nghĩa:** Chỉ mục lưu trữ giá trị đã được tính toán từ một hàm hoặc biểu thức trên cột.
- **Công dụng:** Tối ưu truy vấn sử dụng hàm hoặc biểu thức trong điều kiện WHERE (ví dụ: UPPER(name)).
- **Đặc điểm:**
 - + Hỗ trợ các hàm, biểu thức phức tạp như UPPER(), TO_CHAR(), SUBSTR()...
 - + Có thể áp dụng cho nhiều loại dữ liệu khác nhau.
- **Ưu điểm:**
 - + Tăng tốc độ truy vấn có sử dụng hàm.

– **Nhược điểm:**

- + Làm tăng kích thước chỉ mục nếu sử dụng nhiều biểu thức phức tạp.

– **Cấu trúc:** Lưu giá trị kết quả của hàm/biểu thức cùng với ROWID.

– **Cách sử dụng:**

```
CREATE INDEX idx_name ON table_name(UPPER(column_name));  
ALTER SESSION SET QUERY_REWRITE_ENABLED=TRUE;  
ALTER SESSION SET QUERY_REWRITE_INTEGRITY=TRUSTED;
```

5.1.6. Partitioned Index (Local & Global)

- **Ý nghĩa:** Chỉ mục được chia thành các phần (partition) độc lập, tương ứng với phân vùng của bảng.
- **Công dụng:** Tăng hiệu năng quản lý dữ liệu lớn, truy vấn nhanh trên phân vùng.

5.1.6.1. Local Partitioned Index

– **Đặc điểm:**

- + Mỗi partition của index chỉ liên kết với một partition của bảng.
- + Tự động đồng bộ khi thêm/xóa phân vùng bảng
- + Hỗ trợ partition pruning để tối ưu truy vấn.

– **Ưu điểm:**

- + Dễ bảo trì, có thể rebuild riêng từng phân vùng.
- + Đảm bảo tính độc lập phân vùng (partition independence).

– **Nhược điểm:**

- + Chỉ hỗ trợ phân vùng dựa trên khóa phân vùng của bảng.

– **Cách sử dụng:**

```
-- Local  
CREATE INDEX idx_local ON table_name(column) LOCAL;
```

5.1.6.2. Global Partitioned Index

- **Đặc điểm:**
 - + Index partition có thể tham chiếu đến nhiều table partition.
 - + Phân vùng độc lập với bảng, thường dùng range/hash partitioning.
- **Ưu điểm:**
 - + Linh hoạt trong thiết kế phân vùng.
 - + Phù hợp cho truy vấn không phụ thuộc phân vùng bảng.
- **Nhược điểm:**
 - + Khó bảo trì (phải rebuild khi thay đổi phân vùng bảng).
- **Cách sử dụng:**

```
-- Global Partitioned Index (range-based)
CREATE INDEX idx_global ON table_name(column)
GLOBAL PARTITION BY RANGE (column) (
    PARTITION p1 VALUES LESS THAN (100),
    PARTITION p2 VALUES LESS THAN (MAXVALUE)
);

-- Global Partitioned Index (hash-based)
CREATE INDEX idx_global_hash ON table_name(column)
GLOBAL PARTITION BY HASH (column)
PARTITIONS 4;
```

5.1.7. Invisible Index

- **Ý nghĩa:** Index tồn tại nhưng Oracle sẽ **không sử dụng** trong truy vấn thông thường.
- **Công dụng:** Dùng để kiểm thử, đánh giá hiệu suất index mà không xóa đi.
- **Đặc điểm:**
 - + Có thể chuyển qua lại giữa visible/invisible mà không cần xóa chỉ mục.
 - + Hữu ích khi thử nghiệm hiệu quả chỉ mục trước khi quyết định xóa.
- **Ưu điểm:**
 - + Không ảnh hưởng kế hoạch thực thi truy vấn mặc định.

- + Dễ khôi phục sử dụng nếu cần.
- **Nhược điểm:**
 - + Không tự động được chọn bởi Optimizer.
- **Cấu trúc:** Giống B-tree/bitmap, nhưng được gắn cờ INVISIBLE.
- **Cách sử dụng:**

```
CREATE INDEX idx_name ON table_name(column_name) INVISIBLE;  
ALTER INDEX idx_name VISIBLE;
```

5.1.8. Chỉ mục tổng hợp (Composite/Concatenated Index)

- **Ý nghĩa:** Chỉ mục trên nhiều cột, giúp tối ưu truy vấn kết hợp nhiều điều kiện.
- **Công dụng:** Tìm kiếm nhanh khi truy vấn lọc hoặc sắp xếp trên nhiều cột.
- **Đặc điểm:**
 - + Thứ tự các cột trong chỉ mục ảnh hưởng đến hiệu quả truy vấn.
 - + Thích hợp cho các truy vấn WHERE sử dụng nhiều cột đầu của chỉ mục
- **Ưu điểm:**
 - + Tối ưu cho các truy vấn kết hợp nhiều điều kiện.
- **Nhược điểm:**
 - + Không hiệu quả nếu truy vấn chỉ sử dụng các cột không nằm ở đầu chỉ mục.
- **Cấu trúc:** Lưu trữ giá trị kết hợp của các cột chỉ mục cùng với ROWID.
- **Cách sử dụng:**

```
CREATE INDEX idx_composite ON table(column1, column2);
```

5.1.9. Auto Indexing (Automatic Indexes)

- **Ý nghĩa:** Từ Oracle 19c, hệ quản trị có thể tự động tạo, quản lý và xóa chỉ mục dựa trên thống kê truy vấn thực tế.

- **Công dụng:** Tối ưu tự động hiệu suất truy vấn mà không cần DBA can thiệp thủ công.
- **Đặc điểm:**
 - + Theo dõi workload thực tế.
 - + Chỉ mục tạo ra ban đầu là invisible, sau đó thành visible nếu thực sự hiệu quả.
- **Ưu điểm:**
 - + Tự động hóa tuning.
 - + Có thể rollback nếu không cần.
- **Nhược điểm:**
 - + Chỉ có trên Oracle 19c trở đi và phải bật tính năng AUTO_INDEX.
- **Cách sử dụng:**

```
ALTER SYSTEM SET AUTO_INDEX = ON;
```

5.1.10. Sharded Index

- **Ý nghĩa:** Chỉ mục dùng trong môi trường cơ sở dữ liệu phân mảnh (sharding), tối ưu truy vấn trên bảng được chia nhỏ trên nhiều nút vật lý.
- **Công dụng:** Tối ưu hóa truy cập cục bộ, giảm độ trễ truy vấn trong kiến trúc phân tán.
- **Đặc điểm:**
 - + Phù hợp với kiến trúc phân tán (Oracle Sharding).
 - + Có thể dùng chung với composite sharding key.
- **Ưu điểm:**
 - + Tối ưu hóa truy cập cục bộ.
- **Nhược điểm:**
 - + Chỉ dùng được trong kiến trúc sharded.

5.1.11. Chỉ mục miền ứng dụng (Domain/Application Index)

- **Ý nghĩa:** Chỉ mục tùy chỉnh cho các kiểu dữ liệu đặc biệt hoặc nghiệp vụ riêng.
- **Công dụng:** Tối ưu cho dữ liệu đặc thù như text, spartial, XML, JSON.
- **Đặc điểm:**
 - + Do người dùng hoặc ứng dụng định nghĩa và quản lý.
 - + Chỉ mục miền có thể lưu trữ trong bảng Oracle hoặc ngoài hệ thống.
- **Ưu điểm:**
 - + Tối ưu hóa cho các truy vấn đặc thù ngoài khả năng của chỉ mục chuẩn.
- **Nhược điểm:**
 - + Cần lập trình và quản trị phức tạp hơn.
- **Cấu trúc:** Phụ thuộc vào định nghĩa nghiệp vụ hoặc loại dữ liệu đặc biệt.
- **Cách sử dụng:**

```
CREATE INDEX idx_name ON documents(content) INDEXTYPE IS CTXSYS.CONTEXT;
```

5.1.12. Hash Clustered Index

- **Ý nghĩa:** Dữ liệu được lưu trữ và truy xuất thông qua hàm băm (hash function), không cần index riêng.
- **Công dụng:**
 - + Tối ưu truy vấn đẳng thức (=) trên khóa cluster.
 - + Giảm tranh chấp I/O khi ghi đồng thời
- **Đặc điểm:**
 - + Không cần index riêng: Dữ liệu được định vị trực tiếp qua hash value.
 - + Tham số quan trọng:
 - HASHKEYS: Số lượng giá trị hash tối đa.
 - SIZE: Kích thước trung bình cho mỗi hash key.
- **Ưu điểm:**

- + Truy xuất nhanh với I/O tối thiểu (1 lần đọc).
- + Phân phối dữ liệu đồng đều, tránh "hot blocks".
- **Nhược điểm:**
 - + Không hiệu quả cho truy vấn phạm vi (BETWEEN, >, <).
 - + Yêu cầu thiết lập tham số chính xác để tránh hash collision.
- **Cách sử dụng:**

```
-- Tạo cluster sử dụng HASH
CREATE CLUSTER cluster_name (
    column_name datatype
)
HASHKEYS n
SIZE s;

-- Tạo bảng gắn với CLUSTER
CREATE TABLE table_name (
    column_name datatype,
    ...
)
CLUSTER cluster_name (column_name);

-- Tạo chỉ mục cho CLUSTER
CREATE INDEX cluster_index_name ON CLUSTER cluster_name;
```

5.2. Triển khai các loại chỉ mục trong truy vấn phân tán

Liệt kê danh sách khách hàng đã mua các sản phẩm thuộc thương hiệu Yonex trong năm 2024, hiển thị các thông tin MAKH, HOTEN, NGAYTAO, TENSANPHAM.

```
SELECT * FROM (
    SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
    FROM CN1.HOADON H
    JOIN CN1.CTHD C ON H.MAHD = C.MAHD
    JOIN CN1.SANPHAM SP ON C.MASP = SP.MASP
    JOIN CN1.KHACHHANG KH ON KH.MAKH = H.MAKH
    WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024

    UNION ALL

    SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
```

```

FROM CN2.HOADON@CN2_GiamDoc H
JOIN CN2.CTHD@CN2_GiamDoc C ON H.MAHD = C.MAHD
JOIN CN2.SANPHAM@CN2_GiamDoc SP ON C.MASP = SP.MASP
JOIN CN2.KHACHHANG@CN2_GiamDoc KH ON KH.MAKH = H.MAKH
WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024

UNION ALL

SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
FROM CN3.HOADON@CN3_GiamDoc H
JOIN CN3.CTHD@CN3_GiamDoc C ON H.MAHD = C.MAHD
JOIN CN3.SANPHAM@CN3_GiamDoc SP ON C.MASP = SP.MASP
JOIN CN3.KHACHHANG@CN3_GiamDoc KH ON KH.MAKH = H.MAKH
WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024
);

```

5.2.1. Sử dụng chỉ mục để tối ưu câu truy vấn

EXPLAIN câu truy vấn khi chưa sử dụng chỉ mục:

```

EXPLAIN PLAN FOR
SELECT * FROM (
  SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
  FROM CN1.HOADON H
  JOIN CN1.CTHD C ON H.MAHD = C.MAHD
  JOIN CN1.SANPHAM SP ON C.MASP = SP.MASP
  JOIN CN1.KHACHHANG KH ON KH.MAKH = H.MAKH
  WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024

  UNION ALL

  SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
  FROM CN2.HOADON@CN2_GiamDoc H
  JOIN CN2.CTHD@CN2_GiamDoc C ON H.MAHD = C.MAHD
  JOIN CN2.SANPHAM@CN2_GiamDoc SP ON C.MASP = SP.MASP
  JOIN CN2.KHACHHANG@CN2_GiamDoc KH ON KH.MAKH = H.MAKH
  WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024

  UNION ALL

  SELECT KH.MAKH, KH.HOTEN, H.NGAYTAO, SP.TENSANPHAM
  FROM CN3.HOADON@CN3_GiamDoc H
  JOIN CN3.CTHD@CN3_GiamDoc C ON H.MAHD = C.MAHD
  JOIN CN3.SANPHAM@CN3_GiamDoc SP ON C.MASP = SP.MASP

```

```

JOIN CN3.KHACHHANG@CN3_GiamDoc KH ON KH.MAKH = H.MAKH
WHERE H.ISDELETE = 0 AND SP.THUONGHIEU = 'Yonex' AND EXTRACT(YEAR FROM
H.NGAYTAO) = 2024
);

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY());

```

PLAN_TABLE_OUTPUT VARCHAR2(4000)								
Plan hash value: 2346456307								

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT

0	SELECT STATEMENT		4144	687K	1586 (3)	00:00:01		
1	VIEW		4144	687K	1586 (3)	00:00:01		
2	UNION-ALL							
* 3	HASH JOIN		4144	530K	1586 (3)	00:00:01		
* 4	TABLE ACCESS FULL	SANPHAM	579	36477	7 (0)	00:00:01		
* 5	HASH JOIN		8983	596K	1579 (3)	00:00:01		
* 6	HASH JOIN		2996	152K	569 (4)	00:00:01		
7	TABLE ACCESS FULL	KHACHHANG	1000	25000	5 (0)	00:00:01		
* 8	TABLE ACCESS FULL	HOADON	2996	80892	564 (4)	00:00:01		
9	TABLE ACCESS FULL	CTHD	999K	15M	1003 (2)	00:00:01		
10	REMOTE					CN2_G~ R->S		
11	REMOTE					CN3_G~ R->S		

Hình 5-1. Kết quả sau khi explain câu truy vấn.

- Tạo chỉ mục:

```

-- Function-Based & Composite Index
CREATE INDEX IDX_HD_YEAR_DELETE ON HOADON(ISDELETE, EXTRACT(YEAR FROM
NGAYTAO));

-- B-tree Index trên một cột
CREATE INDEX IDX_HOADON_MAKH ON HOADON(MAKH);

-- Composite B-tree Index (Covering Index)
CREATE INDEX IDX_SP_THUONGHIEU_COVER ON SANPHAM(THUONGHIEU, MASP,
TENSANPHAM);

```

Kết quả EXPLAIN câu truy vấn sau khi sử dụng chỉ mục:

PLAN_TABLE_OUTPUT VARCHAR2(4000)								
Plan hash value: 2904626513								

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT

0	SELECT STATEMENT		4144	687K	1106 (2)	00:00:01		
1	VIEW		4144	687K	1106 (2)	00:00:01		
2	UNION-ALL							
* 3	HASH JOIN		4144	570K	1106 (2)	00:00:01		
* 4	INDEX FAST FULL SCAN	IDX_SP_THUONGHIEU_COVER	579	36477	6 (0)	00:00:01		
* 5	HASH JOIN		8983	684K	1100 (2)	00:00:01		
* 6	HASH JOIN		2996	181K	90 (0)	00:00:01		
7	TABLE ACCESS FULL	KHACHHANG	1000	25000	5 (0)	00:00:01		
8	TABLE ACCESS BY INDEX ROWID BATCHED	HOADON	2996	108K	85 (0)	00:00:01		
* 9	INDEX RANGE SCAN	IDX_HD_YEAR_DELETE	1198		5 (0)	00:00:01		
10	TABLE ACCESS FULL	CTHD	999K	15M	1003 (2)	00:00:01		
11	REMOTE					CN2_G~	R->S	
12	REMOTE					CN3_G~	R->S	

Hình 5-2. Kết quả EXPLAIN câu truy vấn sau khi sử dụng chỉ mục.

Nhận xét: Sau khi sử dụng các chỉ mục phù hợp (bao gồm **function-based index**, **composite index**, và **covering index**), truy vấn thực thi trên môi trường phân tán đã cho thấy hiệu quả rõ rệt. Cụ thể:

- **Chi phí thực thi (Cost)** giảm từ **1586** xuống **1106** (ứng với mức giảm **30.2%**) cho thấy Oracle đã chọn được kế hoạch thực thi hiệu quả hơn nhờ vào chỉ mục.
- Oracle đã sử dụng:
 - + **INDEX FAST FULL SCAN IDX_SP_THUONGHIEU_COVER** trên bảng SANPHAM để lọc theo THUONGHIEU đồng thời truy xuất luôn MASP và TENSANPHAM mà không cần truy cập bảng gốc.
 - + **INDEX RANGE SCAN IDX_HD_YEAR_DELETE** trên bảng HOADON để lọc nhanh theo ISDELETE và năm NGAYTAO, chỉ trả về 1198 dòng thay vì quét toàn bộ bảng.
 - + **TABLE ACCESS BY INDEX ROWID BATCHED** trên bảng HOADON để truy xuất các cột còn lại sau khi đã lọc qua chỉ mục, giảm đáng kể I/O.

- Bảng CTHD vẫn bị quét toàn bộ (FULL TABLE SCAN) do không có điều kiện lọc đặc thù trong truy vấn. Tuy nhiên, vẫn có thể chấp nhận vì Oracle ước lượng việc quét toàn bảng hiệu quả hơn sử dụng chỉ mục trong bối cảnh truy vấn nhiều dòng.
- Duy trì tính phân tán: các **REMOTE operation** cho CN2 và CN3 vẫn được giữ nguyên, đảm bảo tính đúng đắn và đồng nhất của truy vấn phân tán.
- Số dòng kết quả cuối cùng vẫn giữ nguyên (4144 dòng), đảm bảo tính chính xác.
- Mặc dù thời gian ước tính vẫn là 00:00:01, nhưng với chi phí thấp hơn thì truy vấn sẽ thực thi nhanh hơn đáng kể trong thực tế, đặc biệt khi dữ liệu tăng lên.

Kết luận: Việc thiết kế và triển khai chỉ mục phù hợp đã giúp tối ưu hóa đáng kể hiệu năng truy vấn phân tán, chứng minh tầm quan trọng của indexing strategy trong môi trường Oracle distributed database.

6. PHÂN CÔNG CÔNG VIỆC

Bảng 12. Phân công công việc

Tên thành viên	Nội dung công việc	Mức độ hoàn thành	Mức độ đóng góp
Nguyễn Thị Hải Hà	<ul style="list-style-type: none"> – Thiết kế và phân mảnh CSDL, tạo dữ liệu. – Tối ưu hoá câu truy vấn. – Thực hiện và viết báo cáo Indexing in Oracle. – Máy chi nhánh 1 	100%	25%
Nguyễn Thị Thanh Hiếu	<ul style="list-style-type: none"> – Viết 10 câu query. – Viết 4 mức cô lập. – Viết báo cáo. – Chỉnh sửa video BTL1 	100%	25%
Nguyễn Hồng Nhật Long	<ul style="list-style-type: none"> – Viết function và procedure. – Máy chi nhánh 2 	100%	25%
Phạm Đức Mạnh	<ul style="list-style-type: none"> – Tạo dữ liệu. – Viết ràng buộc toàn vẹn. 	100%	25%

	<ul style="list-style-type: none">– Hỗ trợ chỉnh sửa các query và các mức cô lập.– Máy chi nhánh 3		
--	---	--	--

7. TÀI LIỆU THAM KHẢO

- [1] NHOM05_ K2020_CLC, “Bài Tập Lớn 1 - Thiết Kế Cơ Sở Dữ Liệu Phân Tán”, 2022. [Online].
- [2] NHOM15_ K2020_CLC, “Bài Tập Lớn 1 - Thiết Kế Cơ Sở Dữ Liệu Phân Tán”, 2022. [Online].
- [3] Oracle Help Center, “Indexes and Index-Organized Tables”. [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/21/cncpt/indexes-and-index-organized-tables.html>
- [4] Oracle Help Center, “Indexes and Index-Organized Tables”. [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/19/adfns/indexes.html>