

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN I**

**MÔN CƠ SỞ DỮ LIỆU PHÂN TÁN**

**ĐỀ TÀI: HỆ THỐNG KINH DOANH GIÀY**

**IS211.N11.HTCL**

***Giảng viên hướng dẫn:***

**ThS. Thái Bảo Trân**

**KS. Nguyễn Minh Nhựt**

***Nhóm sinh viên thực hiện:***

**Nguyễn Thành Đạt – 19521340**

**Trần Thanh Trung – 19521065**

**Nguyễn Trung Nguyên – 20521678**

**Trần Thanh Phong – 20521750**

*Thành phố Hồ Chí Minh, 12/2022*

# ***LỜI CẢM ƠN***

Nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến với các thầy cô của Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM cũng như quý thầy cô khoa Hệ thống thông tin của trường đã giúp cho chúng em trang bị các kiến thức cơ bản, các kỹ năng thực tế và tạo điều kiện tốt nhất là luôn giúp đỡ, góp ý cho các sinh viên trong quá trình học tập để có thể hoàn thành bài tập lớn của môn.

Đặc biệt nhóm chúng em xin chân thành ơn thầy Nguyễn Minh Nhật đã luôn hết mình hướng dẫn, quan tâm và giúp đỡ khi chúng em gặp khó khăn, bất cập. Thầy luôn truyền đạt những kinh nghiệm tốt nhất cho chúng em.

Mặc khác, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía Thầy/Cô nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

***Nhóm thực hiện.***

[illegible]

## Mục lục

CHƯƠNG 1: THIẾT KẾ CSDL TRÊN MÔI TRƯỜNG RADMIN .....	7
1.1 Thiết kế CSDL phân tán .....	7
1.2 Các câu truy vấn .....	11
1.3 Thực hiện .....	11
2.1 Hàm .....	23
2.2 Thủ tục.....	23
2.2.1 Thủ tục nhập kho .....	23
2.2.2 Thủ tục xem thông tin nhân viên.....	23
2.2.3 Thủ tục bán hàng .....	24
2.3 Ràng buộc toàn vẹn .....	27
CHƯƠNG 3: MỨC CỘ LẬP TRÊN MÔI TRƯỜNG PHẦN TÁN .....	28
3.1 Xem mức cộ lập .....	28
3.2 Các trường hợp mất tính nhất quán.....	28
3.2.1 Lost update .....	28
3.2.2 Dedlock.....	31
3.2.3 Non – repeatable: .....	34
3.2.4 Phantom Read:.....	36
CHƯƠNG 4: TỐI ƯU HÓA CÂU TRUY VẤN .....	39
4.1 Câu truy vấn ban đầu chưa được tối ưu: .....	39
4.2 Phân rã câu truy vấn ban đầu .....	41
4.3 Câu truy vấn đại số quan hệ sau khi tối ưu hóa toàn cục.....	42
4.4 Đánh giá thông qua EXPLAIN plan .....	44
4.5 Lược đồ phân mảnh.....	46
4.6 Cây truy vấn dựa vào lược đồ phân mảnh .....	47
CHƯƠNG 5: CƠ CHẾ NHẬN BẢN TRONG ORACLE.....	48
5.1 Khái niệm.....	48
5.2 Cơ chế nhận bản của RMAN.....	48
5.2.1 Cơ chế nhận bản của Recovery Manager (RMAN) Backup.....	48
5.2.2 Các bước tiến hành.....	48

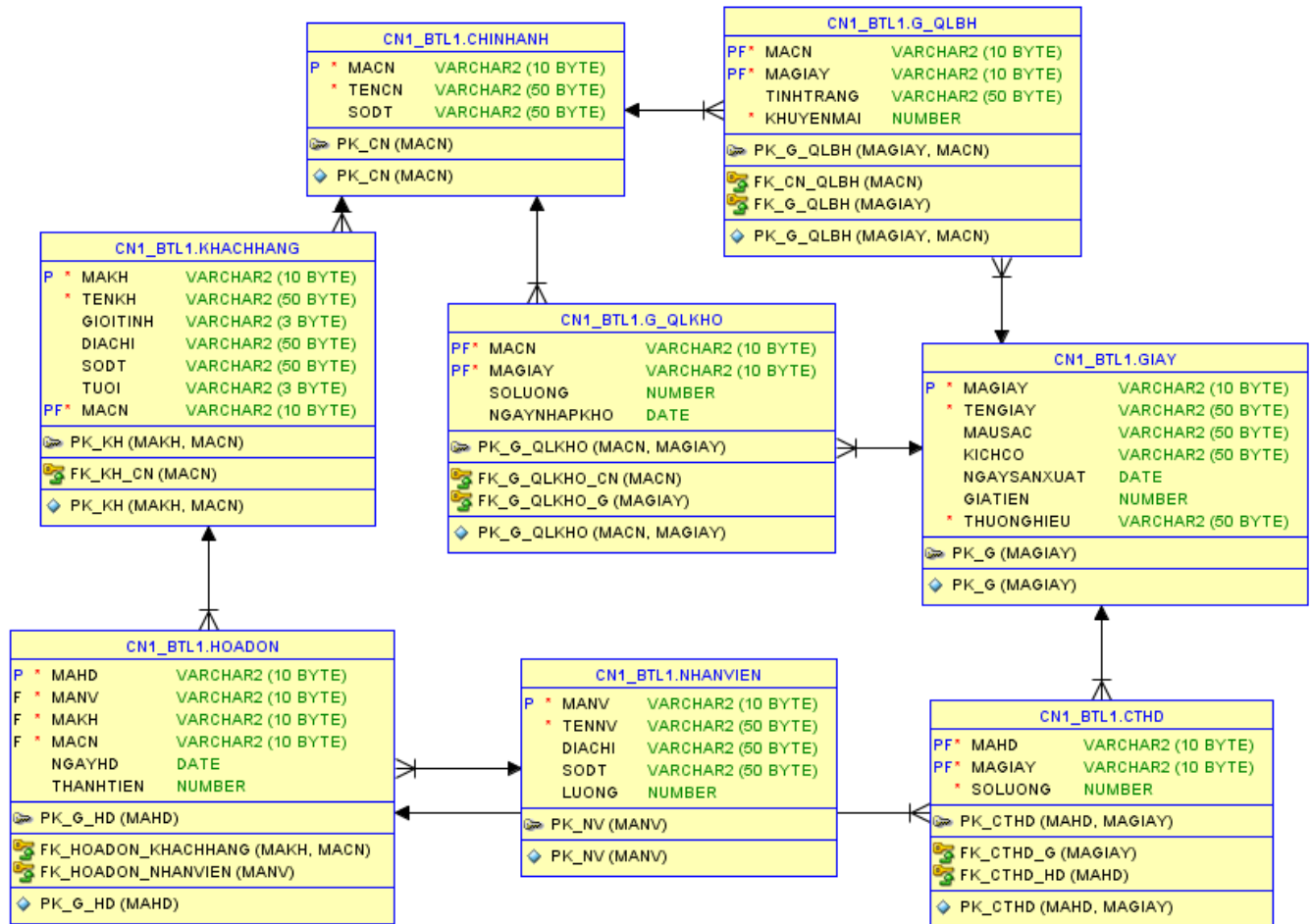
TÀI LIỆU THAM KHẢO..... 49

## Tóm tắt nội dung bài tập lớn

1. Thiết kế CSDL Phân tán trên môi trường máy ảo, radmin
2. Viết hàm, thủ tục, ràng buộc toàn vẹn truy vấn trên môi trường phân tán
3. Demo các mức cô lập (ISOLATION LEVEL) trong môi trường phân tán
4. Thực hiện tối ưu hóa truy vấn trên môi trường phân tán 1 câu truy vấn đơn giản
5. Cơ chế nhân bản trong Oracle

# CHƯƠNG 1: THIẾT KẾ CSDL TRÊN MÔI TRƯỜNG RADMIN

## 1.1 Thiết kế CSDL phân tán



Chi nhánh 1	Chi nhánh 2
<b>CREATE TABLE</b> CN1.GIAY ( MaGiay varchar2(10) NOT NULL, TenGiay varchar2(50) NOT NULL, MauSac varchar2(50), KichCo varchar2(50), NgaySanXuat <b>date</b> , GiaTien number, ThuongHieu varchar2(50) NOT NULL, <b>CONSTRAINT</b> PK_G <b>PRIMARY KEY</b> (MaGiay) );	<b>CREATE TABLE</b> CN2.GIAY ( MaGiay varchar2(10) NOT NULL, TenGiay varchar2(50) NOT NULL, MauSac varchar2(50), KichCo varchar2(50), NgaySanXuat <b>date</b> , GiaTien number, ThuongHieu varchar2(50) NOT NULL, <b>CONSTRAINT</b> PK_G <b>PRIMARY KEY</b> (MaGiay) );
<b>CREATE TABLE</b> CN1.CHINHANH ( MaCN varchar2(10) NOT NULL, TenCN varchar2(50) NOT NULL, SoDT varchar2(50), <b>CONSTRAINT</b> PK_CN <b>PRIMARY KEY</b> (MaCN) );	<b>CREATE TABLE</b> CN2.CHINHANH ( MaCN varchar2(10) NOT NULL, TenCN varchar2(50) NOT NULL, SoDT varchar2(50), <b>CONSTRAINT</b> PK_CN <b>PRIMARY KEY</b> (MaCN) );
<b>CREATE TABLE</b> CN1.KHACHHANG ( MaKH varchar2(10) NOT NULL, TenKH varchar2(50) NOT NULL, GioiTinh varchar2(3), DiaChi varchar2(50), SoDT varchar2(50), Tuoi varchar2(3), MaCN varchar2(10), <b>CONSTRAINT</b> PK_KH <b>PRIMARY KEY</b> (MaKH, MaCN),	<b>CREATE TABLE</b> CN2.KHACHHANG ( MaKH varchar2(10) NOT NULL, TenKH varchar2(50) NOT NULL, GioiTinh varchar2(3), DiaChi varchar2(50), SoDT varchar2(50), Tuoi varchar2(3), MaCN varchar2(10), <b>CONSTRAINT</b> PK_KH <b>PRIMARY KEY</b> (MaKH, MaCN),



<b>CONSTRAINT FK_KH_CN</b> <b>FOREIGN KEY (MaCN) REFERENCES</b> CN1.CHINHANH(MaCN) );	<b>CONSTRAINT FK_KH_CN</b> <b>FOREIGN KEY (MaCN) REFERENCES</b> CN2.CHINHANH(MaCN) );
<b>CREATE TABLE CN1.NHANVIEN</b> ( MaNV varchar2(10) NOT NULL, TenNV varchar2(50) NOT NULL, DiaChi varchar2(50), SoDT varchar2(50), Luong number, <b>CONSTRAINT PK_NV PRIMARY KEY</b> (MaNV) );	<b>CREATE TABLE CN2.NHANVIEN</b> ( MaNV varchar2(10) NOT NULL, TenNV varchar2(50) NOT NULL, DiaChi varchar2(50), SoDT varchar2(50), Luong number, <b>CONSTRAINT PK_NV PRIMARY KEY</b> (MaNV) );
<b>CREATE TABLE CN1.G_QLKHO</b> ( MaCN varchar2(10) NOT NULL, MaGiay varchar2(10) NOT NULL, SoLuong number, NgayNhapKho date, <b>CONSTRAINT PK_G_QLKHO PRIMARY KEY</b> (MaCN, MaGiay), <b>CONSTRAINT FK_G_QLKHO_CN</b> <b>FOREIGN KEY (MaCN) REFERENCES</b> CN1.CHINHANH(MaCN), <b>CONSTRAINT FK_G_QLKHO_G</b> <b>FOREIGN KEY (MaGiay) REFERENCES</b> CN1.GIAY(MaGiay) );	<b>CREATE TABLE CN2.G_QLKHO</b> ( MaCN varchar2(10) NOT NULL, MaGiay varchar2(10) NOT NULL, SoLuong number, NgayNhapKho date, <b>CONSTRAINT PK_G_QLKHO PRIMARY KEY</b> (MaCN, MaGiay), <b>CONSTRAINT FK_G_QLKHO_CN</b> <b>FOREIGN KEY (MaCN) REFERENCES</b> CN2.CHINHANH(MaCN), <b>CONSTRAINT FK_G_QLKHO_G</b> <b>FOREIGN KEY (MaGiay) REFERENCES</b> CN2.GIAY(MaGiay) );
KhuyenMai number NOT NULL, <b>CONSTRAINT PK_G_QLBH PRIMARY KEY</b> (MaGiay, MaCN),	KhuyenMai number NOT NULL, <b>CONSTRAINT PK_G_QLBH PRIMARY KEY</b> (MaGiay, MaCN),

<p>CONSTRAINT FK_CN_QLBH FOREIGN KEY(MaCN) REFERENCES CN1.CHINHANH (MaCN),</p> <p>CONSTRAINT FK_G_QLBH FOREIGN KEY(MaGiay) REFERENCES CN1.GIAY(MaGiay) );</p>	<p>CONSTRAINT FK_CN_QLBH FOREIGN KEY(MaCN) REFERENCES CN2.CHINHANH (MaCN),</p> <p>CONSTRAINT FK_G_QLBH FOREIGN KEY(MaGiay) REFERENCES CN2.GIAY(MaGiay) );</p>
<p>CREATE TABLE CN1.HOADON ( MaHD varchar2(10) NOT NULL, MaNV varchar2(10) NOT NULL, MaKH varchar2(10) NOT NULL, MaCN varchar2(10) NOT NULL, NgayHD date, ThanhTien number, CONSTRAINT PK_G_HD PRIMARY KEY(MaHD), CONSTRAINT FK_HOADON_NHANVIEN FOREIGN KEY (MaNV) REFERENCES CN1.NHANVIEN (MaNV), CONSTRAINT FK_HOADON_KHACHHANG FOREIGN KEY (MaKH) REFERENCES CN1.KHACHHANG (MaKH) );</p>	<p>CREATE TABLE CN2.HOADON ( MaHD varchar2(10) NOT NULL, MaNV varchar2(10) NOT NULL, MaKH varchar2(10) NOT NULL, MaCN varchar2(10) NOT NULL, NgayHD date, ThanhTien number, CONSTRAINT PK_G_HD PRIMARY KEY(MaHD), CONSTRAINT FK_HOADON_NHANVIEN FOREIGN KEY (MaNV) REFERENCES CN2.NHANVIEN (MaNV), CONSTRAINT FK_HOADON_KHACHHANG FOREIGN KEY (MaKH) REFERENCES CN2.KHACHHANG (MaKH) );</p>
<p>CREATE TABLE CN1.CTHD ( MaHD varchar2(10) NOT NULL, MaGiay varchar2(10) NOT NULL, SoLuong number NOT NULL, CONSTRAINT PK_CTHD PRIMARY KEY (MaHD, MaGiay),</p>	<p>CREATE TABLE CN2.CTHD ( MaHD varchar2(10) NOT NULL, MaGiay varchar2(10) NOT NULL, SoLuong number NOT NULL, CONSTRAINT PK_CTHD PRIMARY KEY (MaHD, MaGiay),</p>

<b>CONSTRAINT FK_CTHD_HD FOREIGN KEY (MaHD) REFERENCES CN1.HOADON (MaHD), CONSTRAINT FK_CTHD_G FOREIGN KEY (MaGiay) REFERENCES CN1.GIAY (MaGiay) );</b>	<b>CONSTRAINT FK_CTHD_HD FOREIGN KEY (MaHD) REFERENCES CN2.HOADON (MaHD), CONSTRAINT FK_CTHD_G FOREIGN KEY (MaGiay) REFERENCES CN2.GIAY (MaGiay) );</b>
---	---

## 1.2 Các câu truy vấn

Câu 1: Đưa ra thông tin tất cả giày tại cả 2 chi nhánh với số lượng dưới 10 sắp xếp giảm dần.

Câu 2: Tìm những mẫu giày với tình trạng là “Con hang” và có số lượng lớn hơn 20 còn lại trong kho ở cả 02 chi nhánh.

Câu 3: Tìm những mẫu giày được phân phối tại chi nhánh 2 nhưng không có tại chi nhánh 1 với tình trạng “Con hang”.

Câu 4: Tìm mã khách hàng mua hàng nhiều nhất trong tháng 7 tại chi nhánh 1.

Câu 5: Tìm mã nhân viên ở chi nhánh 2 có số lượng hóa đơn nhiều nhất trong cả hai chi nhánh.

Câu 6: Với mỗi màu sắc giày, liệt kê số lượng mua của khách hàng từ 18 đến 25 tuổi.

Câu 7: Tìm các hóa đơn mua hàng trong tháng 7 với tổng số lượng giày có trên 5 sản phẩm, sắp xếp theo thứ tự giảm dần ở cả 02 chi nhánh.

Câu 8: Tìm mẫu giày có số lượng bán ít nhất trong năm 2021 ở cả 02 chi nhánh.

Câu 9: Tìm mẫu giày được mua bởi cả khách hàng nam và nữ ở cả 02 chi nhánh.

Câu 10: Tìm khách hàng đã mua hết tất cả loại giày có giá lớn hơn 500.000 tại chi nhánh 1.

## 1.3 Thực hiện

Câu 1: Đưa ra thông tin tất cả giày tại cả 2 chi nhánh với số lượng dưới 10 sắp xếp giảm dần.

```

SELECT
    *
FROM
    (
        SELECT
            g.*,
            g_qlkho.soluong
        FROM
            giay g
        JOIN g_qlkho ON g_qlkho.magiay = g.magiay
        WHERE
            soluong < 10
        UNION
        SELECT
            g.*,
            qlk.soluong
        FROM
            cn2_bt11.giay@db1_cn2 g
        JOIN cn2_bt11.g_qlkho@db1_cn2 qlk ON qlk.magiay = g.magiay
        WHERE
            qlk.soluong < 10
    )
ORDER BY
    soluong DESC;

```

Kết quả:

	⚡ MAGIAY	⚡ TENGIAY	⚡ MAUSAC	⚡ KICHCO	⚡ NGAYSANXUAT	⚡ GIATIEN	⚡ THUONGHIEU	⚡ SOLUONG
1	G36	PumaSanji	Yellow	Small	05/10/2020 00:00:00	2090000	Puma	9
2	G21	Zaral	Gold	Small	01/04/2019 00:00:00	999000	Zara	8
3	G48	Balenciaga Sa Pa	White	Lagre	04/07/2022 00:00:00	1190000	Balenciaga	8
4	G27	ZaraBlue	Blue	Medium	02/05/2019 00:00:00	990000	Zara	7
5	G23	Zara3	Red	Large	01/04/2019 00:00:00	390000	Zara	7
6	G31	PumaGold	Yellow	Small	04/09/2018 00:00:00	490000	Puma	7
7	G45	Balenciaga Hoi An	Cyan	Medium	02/04/2021 00:00:00	1790000	Balenciaga	7
8	G38	PumaNami	Pink	Small	05/10/2020 00:00:00	1890000	Puma	6
9	G29	ZaraPro	Grey	Medium	01/01/2019 00:00:00	1290000	Zara	5
10	G43	Balenciaga Da Nang	Blue	Lagre	02/04/2021 00:00:00	1750000	Balenciaga	3
11	G11	Kappa3	Pink	Small	20/11/2022 00:00:00	620000	Kappa	3
12	G50	Balenciaga Viet Nam	Red	Lagre	22/02/2022 00:00:00	999999999	Balenciaga	1

Câu 2: Tìm những mẫu giày với tình trạng là “Con hang” và có số lượng lớn hơn 20 còn lại trong kho ở cả 02 chi nhánh.

```

SELECT
    *
FROM
    (
        SELECT
            g.*,
            qlk.soluong,
            qlk.mactn,
            qlbh.tinhtrang
        FROM
            giay g, g_qlkho qlk, g_qlbh qlbh
        WHERE
            g.magiay = qlk.magiay
            and qlbh.magiay = g.magiay
            and qlbh.tinhtrang = 'Con hang'
            and soluong > 20
        UNION
        SELECT
            g.*,
            qlk.soluong,
            qlk.mactn,
            qlbh.tinhtrang
        FROM
            cn2_bt11.giay@db1_cn2 g, cn2_bt11.g_qlkho@db1_cn2 qlk, cn2_bt11.g_qlbh@db1_cn2 qlbh
        WHERE
            g.magiay = qlk.magiay
            and qlbh.magiay = g.magiay
            and qlbh.tinhtrang = 'Con hang'
            and soluong > 20
    )
ORDER BY
    soluong DESC;

```

Kết quả:

⚡	MAGIAY	TENGIAY	MAUSAC	KICHCO	NGAYSANXUAT	GIATIENT	THUONGHIEU	SOLUONG	MACN	TINHTRANG
1	G04	NIKE4	Black	Small	20/11/2022 00:00:00	704000	Nike	500	CN01	Con hang
2	G47	Balenciaga Ha Long	Gold	Medium	02/04/2021 00:00:00	990000	Balenciaga	307	CN02	Con hang
3	G02	Nike2	White	Medium	20/11/2022 00:00:00	990000	Nike	300	CN01	Con hang
4	G32	PumaS	Black	Small	05/05/2020 00:00:00	1290000	Puma	287	CN02	Con hang
5	G14	Adidas1	Taupe	Medium	20/11/2022 00:00:00	1607700	Adidas	250	CN01	Con hang
6	G42	Balenciaga Ha Noi	Brown	Lagre	02/04/2021 00:00:00	1190000	Balenciaga	250	CN02	Con hang
7	G34	PumaLuffy	Red	Lagre	05/10/2020 00:00:00	2290000	Puma	206	CN02	Con hang
8	G07	Converse2	Blue	Small	20/11/2022 00:00:00	395000	Converse	200	CN01	Con hang
9	G37	PumaChoper	Yellow	Small	05/10/2020 00:00:00	1990000	Puma	157	CN02	Con hang
10	G40	Balenciaga Zgolden	Yellow	Small	02/04/2021 00:00:00	1290000	Balenciaga	150	CN02	Con hang
11	G12	Multi-Bag	Navy	Small	20/11/2022 00:00:00	620000	Kappa	150	CN01	Con hang
12	G09	Kappa1	Orange	Medium	20/11/2022 00:00:00	715500	Kappa	106	CN01	Con hang
13	G20	Adidas8	Cream	Small	20/11/2022 00:00:00	2490000	Adidas	100	CN01	Con hang
14	G46	Balenciaga Phu Quoc	Green	Small	02/04/2021 00:00:00	1290000	Balenciaga	100	CN02	Con hang
15	G44	Balenciaga Sai Gon	Brown	Lagre	02/04/2021 00:00:00	890000	Balenciaga	95	CN02	Con hang
16	G17	Adidas4	Stell Blue	Small	20/11/2022 00:00:00	2410000	Adidas	25	CN01	Con hang

Câu 3: Tìm những mẫu giày được phân phối tại chi nhánh 2 nhưng không có tại chi nhánh 1 với tình trạng “Con hang”.

```

SELECT
    g.*
FROM
    giay
    JOIN cn2_bt11.g_qlbh@db1_cn2    qlbh ON g.magiay = qlbh.magiay
WHERE
    qlbh.tinhtrang = 'Con hang'
MINUS
SELECT
    g.*
FROM
    giay
    JOIN g_qlbh    qlbh ON g.magiay = qlbh.magiay
WHERE
    qlbh.tinhtrang = 'Con hang';

```

Kết quả

	⚡ MAGIAY	⚡ TENGIAY	⚡ MAUSAC	⚡ KICHCO	⚡ NGAYSANXUAT	⚡ GIATIEN	⚡ THUONGHIEU
1	G27	ZaraBlue	Blue	Medium	02/05/2019 00:00:00	990000	Zara
2	G29	ZaraPro	Grey	Medium	01/01/2019 00:00:00	1290000	Zara
3	G32	PumaS	Balck	Small	05/05/2020 00:00:00	1290000	Puma
4	G34	PumaLuffy	Red	Lagre	05/10/2020 00:00:00	2290000	Puma
5	G37	PumaChoper	Yellow	Small	05/10/2020 00:00:00	1990000	Puma
6	G40	BalenciagaZgolden	Yellow	Small	02/04/2021 00:00:00	1290000	Balenciaga
7	G42	Balenciaga Ha Noi	Brown	Lagre	02/04/2021 00:00:00	1190000	Balenciaga
8	G44	Balenciaga Sai Gon	Brown	Lagre	02/04/2021 00:00:00	890000	Balenciaga
9	G46	Balenciaga Phu Quoc	Green	Small	02/04/2021 00:00:00	1290000	Balenciaga
10	G47	Balenciaga Ha Long	Gold	Medium	02/04/2021 00:00:00	990000	Balenciaga
11	G49	Balenciaga Da Lat	Snow	Lagre	07/09/2021 00:00:00	1150000	Balenciaga
12	G50	Balenciaga Viet Nam	Red	Lagre	22/02/2022 00:00:00	999999999	Balenciaga

Câu 4: Tìm mã khách hàng mua hàng nhiều nhất trong tháng 7 tại chi nhánh 1

```

SELECT
    hd.makh,
    COUNT(DISTINCT ct.magiay)
FROM
    hoadon    hd
    JOIN cthd    ct ON hd.mahd = ct.mahd
WHERE
    TO_CHAR(hd.ngayhd, 'MM') = 7
GROUP BY
    hd.makh
HAVING
    COUNT(DISTINCT ct.magiay) >= ALL (
        SELECT
            COUNT(DISTINCT ct.magiay)
        FROM
            hoadon    hd
            JOIN cthd    ct ON hd.mahd = ct.mahd
        WHERE
            TO_CHAR(hd.ngayhd, 'MM') = 7
        GROUP BY
            hd.makh
    );

```

Kết quả:

	MAKH	COUNT(DISTINCTCT.MAGIAY)
1	KH02	2

Câu 5: Tìm mã nhân viên ở chi nhánh 2 có số lượng hóa đơn nhiều nhất trong cả hai chi nhánh.

```
SELECT
    hd.manv,
    COUNT(hd.mahd)
FROM
    (
        SELECT * FROM nhanvien
        UNION
        SELECT * FROM cn2_bt11.nhanvien@db1_cn2
    ) nv
JOIN (SELECT * FROM hoadon
      UNION
      SELECT * FROM cn2_bt11.hoadon@db1_cn2
    ) hd ON nv.manv = hd.manv
GROUP BY
    hd.manv
HAVING
    COUNT(hd.mahd) >= ALL (
        SELECT COUNT(hd.mahd)
        FROM
            (
                SELECT * FROM nhanvien
                UNION
                SELECT * FROM cn2_bt11.nhanvien@db1_cn2
            ) nv
        JOIN
            (
                SELECT * FROM hoadon
                UNION
                SELECT * FROM cn2_bt11.HOADON@db1_cn2
            ) hd ON nv.manv = hd.manv
        GROUP BY hd.manv
    );
```

Kết quả:

	MANV	COUNT(HD.MAHD)
1	NV11	5



Câu 6: Với mỗi màu sắc giày, liệt kê số lượng mua của khách hàng từ 18 đến 25 tuổi.

```
SELECT mausac, SUM(soluong)
FROM giay g,
(
    SELECT * FROM cthd, hoadon hd
    WHERE cthd.mahd = hd.mahd
    UNION
    SELECT * FROM cn2_bt11.cthd@db1_cn2 cthd, cn2_bt11.hoadon@db1_cn2 hd
    WHERE
        cthd.mahd = hd.mahd
) hd
WHERE
    g.magiay = hd.magiay
    AND hd.makh IN (
        SELECT makh FROM
        (
            SELECT makh, tuoi
            FROM khachhang
            UNION
            SELECT makh, tuoi
            FROM cn2_bt11.khachhang@db1_cn2
        )
        WHERE
            tuoi BETWEEN 18 AND 25
    )
GROUP BY mausac;
```

Kết quả:

	MAUSAC	SUM(SOLUONG)
1	Black	39
2	White	20
3	Blue	11
4	Orange	12
5	Pink	6
6	Navy	12
7	Taupe	9
8	Brown	5
9	Light Pink	5
10	Stell Blue	5
11	Cream	1

Câu 7: Tìm các mã hóa đơn mua hàng trong tháng 7 với tổng số lượng giày có trên 5 sản phẩm, sắp xếp theo thứ tự giảm dần ở cả 02 chi nhánh.

```
SELECT a.mahd, a.tsl
FROM
(
    SELECT hd.mahd, SUM(hd.soluong) tsl
    FROM
    (
        SELECT cthd.mahd, ngayhd, magiay, soluong
        FROM cn2_bt11.hoadon@db1_cn2 hd, cn2_bt11.cthd@db1_cn2 cthd
        WHERE hd.mahd = cthd.mahd
        UNION
        SELECT cthd.mahd, ngayhd, magiay, soluong
        FROM hoadon hd, cthd cthd
        WHERE hd.mahd = cthd.mahd
    ) hd
    WHERE TO_CHAR(hd.ngayhd, 'MM') = 7
    GROUP BY hd.mahd
    HAVING SUM(hd.soluong) > 5
) a
ORDER BY a.tsl DESC;
```

Kết quả:

	MAHD	TSL
1	HD07	24
2	HD02	20
3	HD06	10
4	HD04	9
5	HD08	6

Câu 8: Tìm mã giày có số lượng bán ít nhất trong năm 2021 ở cả 02 chi nhánh.

```
SELECT magiay, SUM(tsl)
FROM
(
    SELECT magiay, SUM(soluong) tsl
    FROM cn2_bt11.cthd@db1_cn2 ct1, cn2_bt11.hoadon@db1_cn2 hd1
    WHERE ct1.mahd = hd1.mahd AND TO_CHAR(hd1.ngayhd, 'YYYY') = 2021
    GROUP BY magiay
    UNION
    SELECT magiay, SUM(soluong) tsl
    FROM cthd ct2, hoadon hd2
    WHERE ct2.mahd = hd2.mahd AND TO_CHAR(hd2.ngayhd, 'YYYY') = 2021
    GROUP BY magiay
) ct
GROUP BY MAGIAY
HAVING
SUM(tsl) <= ALL (
    SELECT SUM(tsl)
    FROM
    (
        SELECT magiay, SUM(soluong) tsl
        FROM cn2_bt11.cthd@db1_cn2 ct1, cn2_bt11.hoadon@db1_cn2 hd1
        WHERE ct1.mahd = hd1.mahd AND TO_CHAR(hd1.ngayhd, 'YYYY') = 2021
        GROUP BY magiay
        UNION
        SELECT magiay, SUM(soluong) tsl
        FROM cthd ct2, hoadon hd2
        WHERE ct2.mahd = hd2.mahd AND TO_CHAR(hd2.ngayhd, 'YYYY') = 2021
        GROUP BY magiay
    ) ct
    GROUP BY magiay
);
```

Kết quả:

	MAGIAY	SUM(TSL)
1	G29	1

Câu 9: Tìm mã giày được mua bởi cả khách hàng nam và nữ ở cả 02 chi nhánh.

```
SELECT
    a.magiaiy, b.sl sl_nu, c.sl sl_nam
FROM
    (
        SELECT DISTINCT hd.magiaiy
        FROM
            (
                SELECT * FROM khachhang
                UNION
                SELECT * FROM cn2_bt11.khachhang@db1_cn2
            ) kh
        JOIN (
            SELECT * FROM hoadon JOIN cthd ON hoadon.mahd = cthd.mahd
            UNION
            SELECT *
            FROM cn2_bt11.hoadon@db1_cn2 JOIN cn2_bt11.cthd@db1_cn2 ON hoadon.mahd = cthd.mahd
        ) hd ON kh.makh = hd.makh
        WHERE kh.gioitinh = 'Nu'
    ) INTERSECT
    SELECT DISTINCT hd.magiaiy
    FROM
        (
            SELECT * FROM khachhang
            UNION
            SELECT * FROM cn2_bt11.khachhang@db1_cn2
        ) kh
        JOIN (
            SELECT * FROM hoadon JOIN cthd ON hoadon.mahd = cthd.mahd
            UNION
            SELECT * FROM cn2_bt11.hoadon@db1_cn2 JOIN cn2_bt11.cthd@db1_cn2 ON hoadon.mahd = cthd.mahd
        ) hd ON kh.makh = hd.makh
        WHERE kh.gioitinh = 'Nam'
    ) a,
```

```

(
  SELECT hd.magiay, SUM(hd.soluong) s1
  FROM
    (
      SELECT * FROM khachhang
      UNION
      SELECT * FROM cn2_bt11.khachhang@db1_cn2
    ) kh
  JOIN (
    SELECT * FROM hoadon JOIN cthd ON hoadon.mahd = cthd.mahd
    UNION
    SELECT * FROM cn2_bt11.hoadon@db1_cn2 JOIN cn2_bt11.cthd@db1_cn2 ON hoadon.mahd = cthd.mahd
  ) hd ON kh.makh = hd.makh
  WHERE kh.gioitinh = 'Nu'
  GROUP BY hd.magiay
) b,
(
  SELECT hd.magiay, SUM(hd.soluong) s1
  FROM
    (
      SELECT * FROM khachhang
      UNION
      SELECT * FROM cn2_bt11.khachhang@db1_cn2
    ) kh
  JOIN (
    SELECT * FROM hoadon JOIN cthd ON hoadon.mahd = cthd.mahd
    UNION
    SELECT * FROM cn2_bt11.hoadon@db1_cn2 JOIN cn2_bt11.cthd@db1_cn2 ON hoadon.mahd = cthd.mahd
  ) hd ON kh.makh = hd.makh
  WHERE kh.gioitinh = 'Nam'
  GROUP BY hd.magiay
) c
WHERE a.magiay = b.magiay AND a.magiay = c.magiay;

```

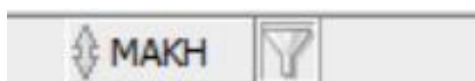
Kết quả:

	MAGIAY	SL_NU	SL_NAM
1	G09	18	13
2	G10	7	2
3	G07	6	4
4	G40	20	9
5	G22	3	5
6	G27	5	2
7	G01	1	16
8	G16	2	3

Câu 10: Tìm mã khách hàng đã mua hết tất cả loại giày có giá lớn hơn 500.000 tại chi nhánh 1.

```
SELECT DISTINCT ( makh )
FROM cnl_bt11.hoadon@db1_cn1 hdl, cnl_bt11.cthd@db1_cn1 ctl, giay gl
WHERE hdl.mahd = ctl.mahd AND ctl.magiay = gl.magiay
AND thuonghieu NOT IN (
SELECT DISTINCT ( thuonghieu )
FROM giay g
WHERE giatien > 500000
AND NOT EXISTS (
SELECT * FROM cnl_bt11.cthd@db1_cn1 ct2, cnl_bt11.hoadon@db1_cn1 hd2
WHERE g.magiay = ct2.magiay AND hd2.mahd = ct2.mahd AND hd2.mahd = hdl.mahd
)
);
```

Kết quả:



(query tại máy 2)

## CHƯƠNG 2: HÀM, THỦ TỤC, RÀNG BUỘC TOÀN VỆN TRÊN MÔI TRƯỜNG PHÂN TÁN

### 2.1 Hàm

Hàm tổng hóa đơn của nhân viên

Tên Function: tonghd

Ý nghĩa, đếm tổng các hóa đơn được thực hiện bởi một nhân viên mới mã nhân viên v\_manv

```
CREATE OR REPLACE FUNCTION tonghd (  
    v_manv IN VARCHAR2  
) RETURN NUMBER IS  
    tong1    NUMBER := 0;  
    tong2    NUMBER := 0;  
BEGIN  
    SELECT COUNT(mahd) INTO tong1 FROM hoadon  
    WHERE manv = v_manv;  
    SELECT COUNT(mahd) INTO tong2 FROM cn2_bt11.hoadon@db1_cn2  
    WHERE manv = v_manv;  
    RETURN tong1 + tong2;  
END;
```

### 2.2 Thủ tục

#### 2.2.1 Thủ tục nhập kho

Tên Procedure: nhapkho

Ý nghĩa, nhập giấy vào kho, sử dụng mã giấy và số lượng giấy nhập vào

```
CREATE OR REPLACE PROCEDURE nhapkho ( v_magiaiy IN VARCHAR2, v_soluong IN NUMBER )  
AS  
    dem INT;  
BEGIN  
    SELECT COUNT(magiaiy) INTO dem FROM g_qlkho  
    WHERE magiaiy = v_magiaiy;  
    IF ( dem > 0 ) THEN  
        UPDATE g_qlkho  
        SET soluong = soluong + v_soluong, ngaynhapkho = current_date  
        WHERE magiaiy = v_magiaiy;  
    ELSE  
        INSERT INTO g_qlkho VALUES ( 'CN01', v_magiaiy, v_soluong, current_date );  
        INSERT INTO g_qlbh VALUES ( 'CN01', v_magiaiy, 'Con Hang', 0 );  
    END IF;  
    COMMIT;  
END;
```

#### 2.2.2 Thủ tục xem thông tin nhân viên

Tên Procedure: thongtinnv

Ý nghĩa: Xem thông tin của một nhân viên dựa trên mã nhân viên

```
CREATE OR REPLACE PROCEDURE thongtinnv (  
    v_manv IN VARCHAR2  
) AS  
  
    tennv      nhanvien.tennv%TYPE;  
    diachi     nhanvien.diachi%TYPE;  
    sodt       nhanvien.sodt%TYPE;  
BEGIN  
    SELECT  
        tennv,  
        diachi,  
        sodt  
    INTO  
        tennv,  
        diachi,  
        sodt  
    FROM  
        nhanvien  
    WHERE  
        manv = v_manv;  
  
    dbms_output.put_line('-----');  
    dbms_output.put_line('Ten NV: ' || tennv);  
    dbms_output.put_line('Dia Chi: ' || diachi);  
    dbms_output.put_line('So DT: ' || sodt);  
    dbms_output.put_line('-----');  
    dbms_output.put_line('Tong so HD: ' || tonghd(v_Manv));  
END;
```

### 2.2.3 Thủ tục bán hàng

Tên Procedure: banhang

Ý nghĩa: Bán giày cho khách hàng, sử dụng mã hóa đơn, mã khách hàng, mã giày và số lượng để bán giày cho khách hàng.



```

CREATE OR REPLACE PROCEDURE banhang ( v_mahd IN VARCHAR2, v_makh IN VARCHAR2, v_magiaiy IN VARCHAR2, v_soluong IN NUMBER)
AS
    dem          NUMBER;
    soluong       NUMBER;
    tim           NUMBER;
    dem_mahd      NUMBER;
    cn            chinhanh.mactn%TYPE;
BEGIN
-- Tìm xem Mã HD nằm ở CN1 hay CN2
    SELECT COUNT(mahd) INTO dem_mahd FROM hoadon
    WHERE mahd = v_mahd AND mactn = 'CN01' AND makh = v_makh;
    IF ( dem_mahd > 0 ) THEN
-- Mã HD ở CN1
-- Kiểm tra số lượng hàng trong kho CN1 để xuất kho bán hàng
        SELECT COUNT(magiaiy) INTO dem FROM g_qlkho
        WHERE magiaiy = v_magiaiy;
        SELECT soluong INTO soluong FROM g_qlkho
        WHERE magiaiy = v_magiaiy;
        IF ( ( dem > 0 ) AND ( soluong - v_soluong >= 0 ) ) THEN
-- Kiểm tra hoá đơn đã mua giấy này chưa, nếu rồi thì cập nhật số lượng, chưa thì insert vào CTHD
            SELECT COUNT(magiaiy) INTO tim FROM cthd
            WHERE mahd = v_mahd AND magiaiy = v_magiaiy;
            IF ( tim > 0 ) THEN
                UPDATE cthd
                SET soluong = soluong + v_soluong
                WHERE mahd = v_mahd AND magiaiy = v_magiaiy;
            ELSE
                INSERT INTO cthd VALUES ( v_mahd, v_magiaiy, v_soluong );
            END IF;
        ELSE
-- Thông báo không đủ hàng
            raise_application_error(-20000, 'Không đủ hàng để xuất kho');
            ROLLBACK;
        END IF;
    ELSE
--Mã HD ở CN2
        SELECT COUNT(mahd) INTO dem_mahd FROM cn2_bt11.hoadon@db1_cn2
        WHERE mahd = v_mahd AND mactn = 'CN01' AND makh = v_makh;
        IF ( dem_mahd > 0 ) THEN
-- Mã HD ở CN2
-- Kiểm tra số lượng hàng trong kho CN1 để xuất kho bán hàng
            SELECT COUNT(magiaiy) INTO dem
            FROM g_qlkho
            WHERE magiaiy = v_magiaiy;
            SELECT soluong INTO soluong FROM g_qlkho
            WHERE magiaiy = v_magiaiy;
            IF ( ( dem > 0 ) AND ( soluong - v_soluong >= 0 ) ) THEN
-- Kiểm tra hoá đơn đã mua túi xách này chưa, nếu rồi thì cập nhật số lượng, chưa thì insert vào CTHD
                SELECT COUNT(magiaiy) INTO tim FROM cn2_bt11.cthd@db1_cn2
                WHERE mahd = v_mahd AND magiaiy = v_magiaiy;
                IF ( tim > 0 ) THEN
                    UPDATE cn2_bt11.cthd@db1_cn2
                    SET soluong = soluong + v_soluong
                    WHERE mahd = v_mahd AND magiaiy = v_magiaiy;
                ELSE
                    INSERT INTO cn2_bt11.cthd@db1_cn2 VALUES ( v_mahd, v_magiaiy, v_soluong);
                END IF;
            ELSE
                INSERT INTO cn2_bt11.cthd@db1_cn2 VALUES ( v_mahd, v_magiaiy, v_soluong);
            END IF;
        ELSE

```

```
-- Thông báo không đủ hàng
        raise_application_error(-20000, 'Khong du hang de xuat kho');
        ROLLBACK;
    END IF;
ELSE
        raise_application_error(-20000, 'Khong tim thay hoa don');
    END IF;
END IF;
COMMIT;
END;
```

### 2.3 Ràng buộc toàn vẹn

Giới tính của khách hàng chỉ có thể là 'Nam' hoặc 'Nữ'

Nội dung:

$\forall n \in KHACHHANG (n.GIOITINH \in \{ 'Nam', 'Nu' \})$

Bối cảnh: quan hệ KHACHHANG

Bảng tầm ảnh hưởng:

	THÊM	XÓA	SỬA
KHACHHANG	+	-	+(GIOITINH)

Code:

```
-- RBTV  
ALTER TABLE KHACHHANG  
ADD CONSTRAINT ck_kh  
CHECK (GIOITINH='Nam' OR GIOITINH='Nu');
```

## CHƯƠNG 3: MỨC CÔ LẬP TRÊN MÔI TRƯỜNG PHÂN TÁN

### 3.1 Xem mức cô lập

```
-- MỨC CÔ LẬP
DECLARE
    trans_id VARCHAR2(100);
BEGIN
    trans_id := dbms_transaction.local_transaction_id(true);
END;

--XEM MỨC CÔ LẬP
SELECT s.sid, s.serial#,
       CASE bitand(t.flag, power(2, 28))
         WHEN 0 THEN
           'READ COMMITTED'
         ELSE
           'SERIALIZABLE'
       END AS isolation_level
FROM   v$transaction  t
       JOIN v$session s ON t.addr = s.taddr
              AND s.sid = sys_context('USERENV', 'SID');
```

### 3.2 Các trường hợp mất tính nhất quán

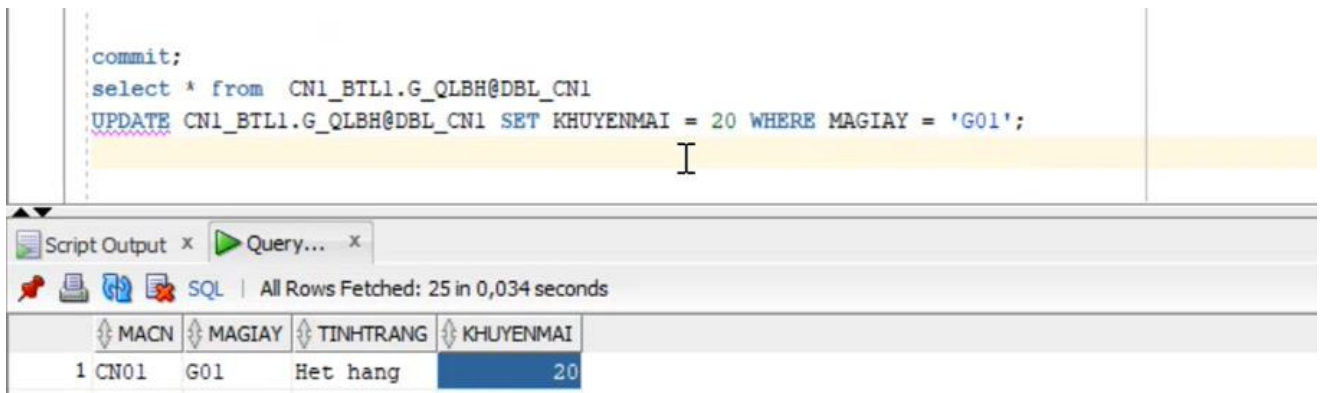
#### 3.2.1 Lost update

- Xảy ra khi hai giao dịch khác nhau đang cố gắng cập nhật cùng một cột trên cùng một hàng trong cơ sở dữ liệu cùng một lúc. Thông thường, một giao dịch cập nhật một cột cụ thể trong một hàng cụ thể, trong khi một giao dịch khác bắt đầu ngay sau đó không thấy bản cập nhật này trước khi cập nhật cùng một giá trị. Kết quả của giao dịch đầu tiên sau đó là "bị mất", vì nó chỉ đơn giản là bị ghi đè bởi giao dịch thứ hai.

- Giả sử, máy 1 xem giày có mã giày 'G01', sau đó cập nhật khuyến mãi của giày từ 0 thành 10% nhưng chưa commit, máy 2 thực hiện cập nhật giày 'G01' với khuyến mãi là 20% và commit, máy 1 bị treo, máy 2 commit trả lại khóa, máy 2 cập nhật thành công, tại máy 1 xem lại khách hàng KH01 thì thấy kết quả cập nhật đã bị mất thay vào đó là kết quả cập nhật tại máy 2.

Máy 1	Máy 2	Mô tả
SELECT * FROM G_QLBH  WHERE MAGIAY = 'G01';  KHUYENMAI = 0;	No action	Máy 1 thực hiện câu lệnh truy vấn xem giày 'G01' thì thấy KHUYENMAI = 0
UPDATE G_QLBH  SET KHUYENMAI = 10	No action	Máy 1 thực hiện cập nhật cập nhật KHUYENMAI của giày

WHERE MAGIAY = 'G01'; <i>1 row updated.</i>		'G01' thành 10
No action	UPDATE CN1.G_QLBH@DBL_CN1 SET KHUYENMAI = 20 WHERE MAGIAY = 'G01';	Máy 2 cập nhật KHUYENMAI tại máy 1 và treo do máy 1 đang cầm khóa.
COMMIT; <i>Commit completed.</i>	No action	Máy 1 commit, thực hiện thành công và kết thúc.
No action	<i>1 row updated</i>	Máy 2 thực hiện thành công
No action	COMMIT; <i>Commit completed.</i>	Máy 2 thực commit và kết thúc
SELECT * FROM G_QLBH WHERE MAGIAY = 'G01'; <i>KHUYENMAI = 20;</i>	No action	Máy 1 thực hiện truy vấn thông tin Giày 'G01' nhưng giá trị không đúng với giá trị đã cập nhật.



→ **Lost Update đã xảy ra.**

Nguyên nhân và giải pháp:

- Vấn đề: Khi Máy 1 cập nhật số dư cho giày 'G01' nhưng sau khi xem lại thì không đúng dữ liệu.
- Nguyên nhân: máy 2 cập nhật dữ liệu trước khi máy 1 tiến hành commit hoặc rollback vì vậy dữ liệu hiển thị tại máy 2 commit sau khi máy 2 update nó sẽ chờ khóa của máy 1 và không thể thực hiện hành động khác nên không thể commit, trước khi máy 1 commit.

- Giải pháp: đổi mức cô lập ở máy 1 thành **SERIALIZABLE** thay vì **READ COMMITTED** trước khi thực thi.

- Xử lý Lost Update bằng Serializable.

Máy 1	Máy 2	Mô tả
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  <i>Transaction ISOLATION succeeded</i>	No action	Máy 1 thiết lập mức cô lập SERIALIZABLE
SELECT * FROM G_QLBH  WHERE MAGIAY = 'G01';  <i>KHUYENMAI = 0;</i>	No action	Máy 1 thực hiện câu lệnh truy vấn xem giày 'G01' thì thấy KHUYENMAI = 0
UPDATE G_QLBH  SET KHUYENMAI = 10  WHERE MAGIAY = 'G01';  <i>1 row updated.</i>	No action	Máy 1 thực hiện cập nhật cập nhật KHUYENMAI của giày 'G01' thành 10
No action	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  <i>Transaction ISOLATION succeeded</i>	Máy 2 thiết lập mức cô lập SERIALIZABLE
No action	UPDATE CN1.G_QLBH@DBL_CN1  SET KHUYENMAI = 20  WHERE MAGIAY = 'G01';	Máy 2 cập nhật KHUYENMAI tại máy 1 và treo do máy 1 đang giữ khóa
COMMIT;  <i>Commit completed.</i>	No action	Máy 1 commit, thực hiện thành công và kết thúc.
No action	<i>ORA-08177: can't serialize access for this transaction</i>	Câu lệnh COMMIT đã kết thúc phiên làm việc của máy 1 và thay đổi dữ liệu trong bảng giày, nên máy 2 không thực hiện thành công với lỗi

		'ORA-08177'.
No action	COMMIT;  <i>Commit completed.</i>	Máy 2 thực commit và kết thúc
SELECT * FROM G_QLBH  WHERE MAGIAY = 'G01';  <i>KHUYENMAI = 10;</i>	No action	Máy 1 thực hiện truy vấn thông tin của giày 'G01' và thấy khuyến mãi mới đã cập nhật đúng với mong muốn.

### 3.2.2 Dedlock

- Là một tình huống không mong muốn trong đó hai hoặc nhiều giao dịch đang chờ đợi nhau vô thời hạn để từ bỏ khóa

- Giả sử máy một thực hiện cập nhật dữ liệu cho khách hàng 'KH01', kết quả thành công nhưng chưa commit, và cầm khóa. Máy 2 thực hiện cập nhật dữ liệu cho khách hàng 'KH02', kết quả thành công và chưa commit, và cầm khóa. Tại máy 1 thực hiện cập nhật khách hàng 'KH02', điều này không đc do máy 2 đang cầm khóa khách hàng 'KH02' và máy 1 bị treo, tại máy 2 thực hiện cập nhật khách hàng 'KH01', điều này không được do máy 1 đang cầm khóa KH01 và máy 2 bị treo. Sau một khoảng thời gian, hệ thống call back lại máy 1, khôi phục máy 1, máy 1 nhả khóa. Máy 2 nhận khóa và cập nhật KH01 thành công, tại đây ta thấy deadlock xảy ra và dữ liệu cập nhật tại máy 1 không thành công.

Máy 1	Máy 2	Mô tả
UPDATE KHACHHANG  SET DIACHI = 'Quan 2, TP HCM'  WHERE MAKH = 'KH01';  <i>1 row updated.</i>	No action	Máy 1 cập nhật cho KH01 với địa chỉ là 'Quan 2, TP HCM'
No action	UPDATE CN1_BT1.KHACHHANG@DBL_CN1  SET DIACHI = 'Quan Go Vap, TP HCM'  WHERE MAKH = 'KH02';  <i>1 row updated.</i>	Máy 2 cập nhật cho KH02 với tên mới là 'Quan Go Vap, TP HCM'
UPDATE KHACHHANG  SET DIACHI = 'Quan 3, TP HCM'	No action	Cập nhật cho khách hàng có mã là KH02, máy 1 phải xin khóa ghi rồi tiến hành cập nhật.

WHERE MAKH = 'KH02';		Nhưng máy 2 đang giữ khóa phải chờ máy 2 nhả khóa.
(Treo)	UPDATE CN1_BT1.KHACHHANG@DBL_CN1  SET DIACHI = 'TP Da Lat, Lam Dong'  WHERE MAKH = 'KH01';	Cập nhật tên cho khách hàng có mã là KH01, máy 2 phải xin khóa ghi rồi tiến hành cập nhật. Nhưng máy 1 đang giữ khóa phải chờ máy 1 nhả khóa.
SQL Error: ORA-00060: deadlock detected while waiting for resource 00060. 00000 - "deadlock detected while waiting for resource"	(Treo)	Máy 1, 2 chờ khóa của nhau nên DBMS bị treo. Sau 1 khoảng thời gian DBMS tạo ra exception
<i>Rollback completed</i>	No action	Giải phóng máy 1.
No action	<i>1 Row updated</i>	Máy 2 xin được khóa và hoàn thành cập nhật thông tin.

→ **Deadlock đã xảy ra.**

Nguyên nhân và giải pháp

- Vấn đề xảy ra : Máy 2 chờ máy 2 giải phóng khóa ghi, máy 1 lại chờ máy 2 giải phóng khóa ghi. Kết quả dẫn đến hệ quản trị báo lỗi và hủy giao tác máy 1.

- Giải pháp : Trong Oracle có cơ chế hỗ trợ giải quyết deadlock cho nên sau một thời gian nó sẽ tự động break cái update chờ đầu tiên. Cho nên sau đó máy 1 rollback thì tại máy 1 được updated thành công.

Máy 1	Máy 2	Mô tả
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  <i>Transaction ISOLATION succeeded</i>	No action	Máy 1 thiết lập mức cô lập SERIALIZABLE
No action	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  <i>Transaction ISOLATION succeeded</i>	Máy 2 thiết lập mức cô lập SERIALIZABLE



UPDATE KHACHHANG  SET DIACHI = 'Quan 2, TP HCM'  WHERE MAKH = 'KH01';  <i>1 row updated.</i>	No action	Máy 1 cập nhật cho KH01 với địa chỉ là 'Quan 2, TP HCM'
No action	UPDATE CN1_BTL1.KHACHHANG@DBL_CN1  SET DIACHI = 'Quan Go Vap, TP HCM'  WHERE MAKH = 'KH02';  <i>1 row updated.</i>	Máy 2 cập nhật cho KH02 với tên mới là 'Quan Go Vap, TP HCM'
UPDATE KHACHHANG  SET DIACHI = 'Quan 3, TP HCM'  WHERE MAKH = 'KH02';	No action	Cập nhật cho khách hàng có mã là KH02, máy 1 phải xin khóa ghi rồi tiến hành cập nhật. Nhưng máy 2 đang giữ khóa phải chờ máy 2 nhả khóa.
(Treo)	UPDATE CN1_BTL1.KHACHHANG@DBL_CN1  SET DIACHI = 'TP Da Lat, Lam Dong'  WHERE MAKH = 'KH01';	Cập nhật tên cho khách hàng có mã là KH01, máy 2 phải xin khóa ghi rồi tiến hành cập nhật. Nhưng máy 1 đang giữ khóa phải chờ máy 1 nhả khóa.
SQL Error: ORA-00060: deadlock detected while waiting for resource 00060. 00000 - "deadlock detected while waiting for resource"	(Treo)	Máy 1, 2 chờ khóa của nhau nên DBMS bị treo. Sau 1 khoảng thời gian DBMS tạo ra exception
<i>Rollback completed</i>	<i>Error</i>	Giải phóng máy 1. Máy 2 thông báo không cập nhật được.
No action	COMMIT;  <i>Commit completed.</i>	Máy 2 commit, hoàn thành giao tác. Hủy giao tác tại máy 2 cập nhật lên máy 1.

COMMIT;  <i>Commit completed.</i>	<i>No action</i>	Máy 1 commit, hoàn thành giao tác. Hủy giao tác tại máy 1 cập nhật lên máy 2.
---	------------------	---

### 3.2.3 Non – repeatable:

- Tình trạng này xảy ra khi một giao tác T1 vừa thực hiện xong thao tác đọc trên một đơn vị dữ liệu (nhưng chưa commit) thì giao tác khác (T2) lại thay đổi (ghi) trên đơn vị dữ liệu này. Điều này làm cho lần đọc sau đó của T1 không còn nhìn thấy dữ liệu ban đầu nữa.

- Giả sử, tại máy 1 thực hiện lệnh xem các khách hàng và thấy số dư của KH01 là 100000, tại máy 2 nạp vào tài khoản KH01 200000 sau đó commit, máy 2 thực hiện rút 100000 ra khỏi tài khoản, nhưng sau khi xem lại tại tài khoản 2 sau khi rút tiền hiện giờ là 200000

Máy 1	Máy 2	Mô tả
SELECT *  FROM G_QLBH  WHERE MAGIAY = 'G02';  <i>KHUYENMAI = 10</i>	No action	Máy 1 thực hiện câu lệnh truy vấn thấy khuyến mãi của giày 'G02' là 10%
No action	UPDATE CN1_BTL1.G_QLBH@DBL_CN1  SET KHUYENMAI = 20 WHERE  MAGIAY = 'G02';	Máy 2 sử dụng mức cô thực hiện lệnh đổi khuyến mãi của giày 'G02' thành 20.
No action	COMMIT;  <i>Commit completed.</i>	Máy 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.
SELECT *  FROM G_QLBH  WHERE MAGIAY = 'G02';  <i>KHUYENMAI = 20</i>	No action	Máy 1 thực hiện câu lệnh truy vấn thấy khuyến mãi của giày 'G02' là 20%

→ **Unrepeatable Read đã xảy ra**

Nguyên nhân và giải pháp:

- Vấn đề: máy 1 thực hiện truy vấn thông tin, ngay lúc này máy 2 cập nhật giá dịch vụ tương ứng, sau đó máy 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.

- Nguyên nhân: vì mức cô lập của Máy 1 là **READ COMMITTED** nên mỗi lần truy vấn trên cùng một đơn vị dữ liệu sẽ đọc lại từ cơ sở dữ liệu (cơ sở dữ liệu lúc này có thể đã bị thay đổi bởi máy khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.

- Giải pháp: đổi mức cô lập ở Máy 1 thành **SERIALIZABLE** thay vì **READ COMMITTED** trước khi thực thi.

- Xử lý Non – repeatable Read bằng Serializable.

Máy 1	Máy 2	Mô tả
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  <i>Transaction ISOLATION succeeded</i>	No action	Máy 1 thiết lập mức cô lập Serializable.
SELECT *  FROM G_QLBH  WHERE MAGIAY = 'G02';  <i>KHUYENMAI = 10</i>	No action	Máy 1 thực hiện câu lệnh truy vấn thấy khuyến mãi của giày 'G02' là 10%
No action	UPDATE CN1_BTL1.G_QLBH@DBL_CN1  SET KHUYENMAI = 20 WHERE  MAGIAY = 'G02';	Máy 2 sử dụng mức cô lập thực hiện lệnh đổi khuyến mãi của giày 'G02' thành 20.
No action	COMMIT;  <i>Commit completed.</i>	Máy 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.
SELECT *  FROM G_QLBH  WHERE MAGIAY = 'G02';  <i>KHUYENMAI = 10</i>	No action	Máy 1 thực hiện câu lệnh truy vấn thấy khuyến mãi của giày vẫn là 10%

COMMIT;  <i>Commit completed.</i>		Máy 1 kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.
---	--	--

### 3.2.4 Phantom Read:

- Là tình trạng mà một giao tác đang thao tác trên một tập dữ liệu nhưng giao tác khác lại chèn thêm các dòng dữ liệu vào tập dữ liệu mà giao tác kia quan tâm dẫn đến 2 queries giống hệt nhau được thực hiện nhưng kết quả trả về lại khác nhau.

- Giả sử: Lần đầu tiên máy xem danh sách khách hàng thì cả máy 1 và máy 2 đều thấy có 20 hóa đơn trong hóa đơn HOADON. Sau đó máy 1 tạo hóa đơn với nhân viên 'NV01' và khách hàng 'KH01'. Tại máy 2 thực hiện lại việc xem danh sách thì thấy có 21 hóa đơn.

Máy 1	Máy 2	Mô tả
SELECT *  FROM HOADON	No action	Máy 1 xem danh sách hóa đơn và thấy có 20 hóa đơn.
No action	SELECT *  FROM CN1_BTL1.HOADON@DBL_CN1	Máy 2 xem danh sách hóa đơn và thấy có 20 hóa đơn.
BEGIN  TAOHD('NV01','KH01');  END;  <i>PL/SQL procedure successfully completed</i>	No action	Máy 1 tạo hóa đơn với nhân viên 'NV01' và khách hàng 'KH01'.
COMMIT;  <i>Commit completed.</i>	No action	Máy 1 thực hiện thành công và kết thúc phiên làm việc.
No action	SELECT *  FROM CN1_BTL1.HOADON@DBL_CN1	Máy 2 xem danh sách hóa đơn và thấy có 21 hóa đơn

→ **Phantom Read đã xảy ra.**

Nguyên nhân và giải pháp:

- Vấn đề: máy 1 thực hiện truy vấn thông tin tình trạng phòng trống, ngay lúc này máy 2 thêm phòng,

sau đó máy 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.

- Nguyên nhân: vì mức cô lập của máy 1 là **READ COMMITTED** nên mỗi lần truy vấn trên cùng một đơn vị dữ liệu sẽ đọc lại từ cơ sở dữ liệu (cơ sở dữ liệu lúc này có thể đã bị thay đổi bởi máy khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.

- Giải pháp: đổi mức cô lập ở máy 1 thành **SERIALIZABLE** thay vì **READ COMMITTED**.

- Xử lý Phantom Read bằng **Serializable**

Máy 1	Máy 2	Mô tả
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  <i>Transaction ISOLATION succeeded</i>	No action	Máy 1 thiết lập mức cô lập Serializable.
	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  <i>Transaction ISOLATION succeeded</i>	
SELECT *  FROM HOADON	No action	Máy 1 xem danh sách hóa đơn và thấy có 20 hóa đơn.
No action	SELECT *  FROM CN1_BTL1.HOADON@DBL_CN1	Máy 2 xem danh sách hóa đơn và thấy có 20 hóa đơn.
BEGIN  TAOHD('NV01','KH01');  END;  <i>PL/SQL procedure successfully completed</i>	No action	Máy 1 tạo hóa đơn với nhân viên 'NV01' và khách hàng 'KH01'.
COMMIT;  <i>Commit completed.</i>	No action	Máy 1 thực hiện thành công và kết thúc phiên làm việc.
No action	SELECT *  FROM CN1_BTL1.HOADON@DBL_CN1	Máy 2 xem danh sách hóa đơn và thấy có 20 hóa đơn

No action	COMMIT;  <i>Commit completed.</i>	Máy 2 commit và kết thúc phiên làm việc
-----------	---	---

## CHƯƠNG 4: TỐI ƯU HÓA CÂU TRUY VẤN

Cho biết những nhân viên(Mã nhân viên, Họ tên, SODT) và tên chi nhánh của kho có lương trên 5 triệu đã những đôi giày có giá từ 1 triệu cho các khách hàng Nam

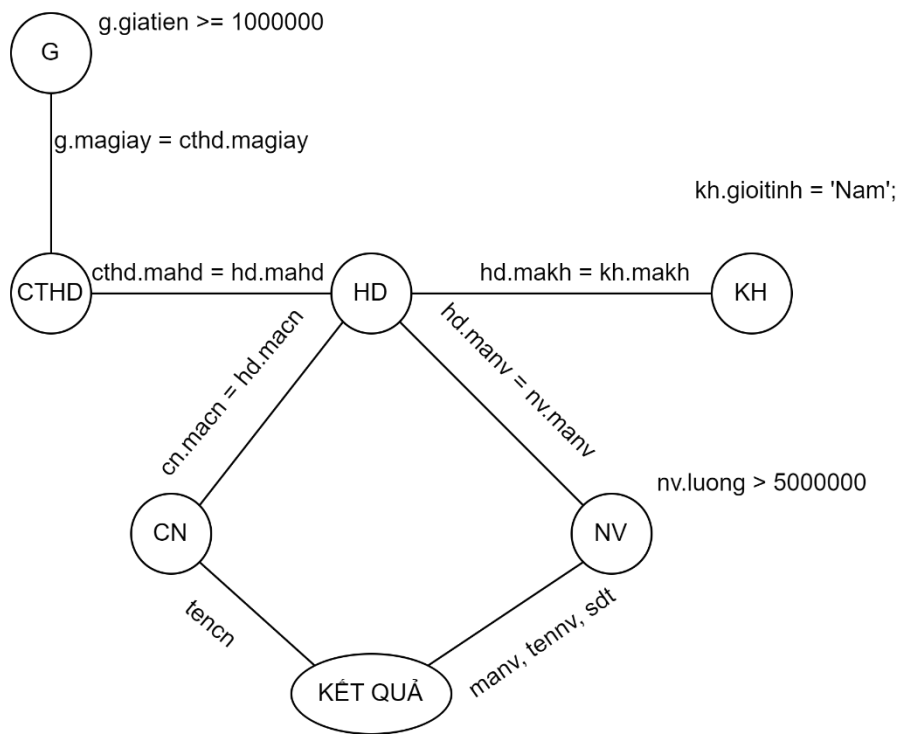
### 4.1 Câu truy vấn ban đầu chưa được tối ưu:

```
select nv.manv, nv.tennv, nv.sodt, cn.tencn
FROM
    nhanvien      nv,
    hoadon        hd,
    cthd,
    giay          g,
    khachhang     kh,
    chinhanh      cn
WHERE
    nv.manv = hd.manv
    AND hd.mahd = cthd.mahd
    AND g.magiay = cthd.magiay
    AND kh.makh = hd.makh
    AND cn.macn = hd.macn
    AND nv.luong > 5000000
    AND g.giatien >= 1000000
    AND kh.gioitinh = 'Nam';
```

Kết quả

	MANV	TENNV	SODT	TENCN
1	NV07	Nguyen Dang Khoa	0949821299	Quan 9, TPHCM
2	NV19	Hoang Tran Bao Long	09528392386	Quan 9, TPHCM
3	NV05	Tran Dang An Nhen	0944924928	Quan 9, TPHCM

Kiểm tra ngữ nghĩa của truy vấn

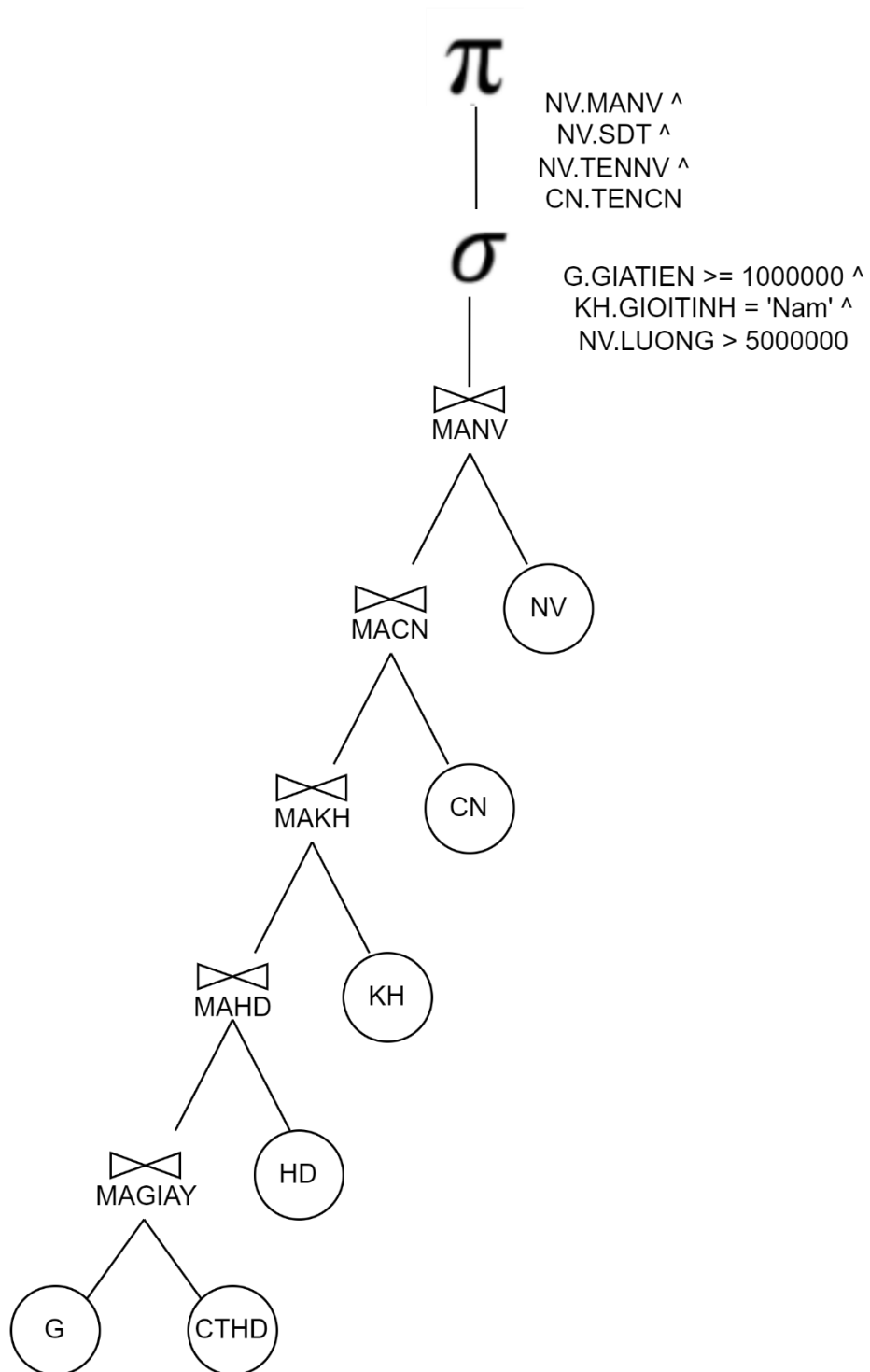



---

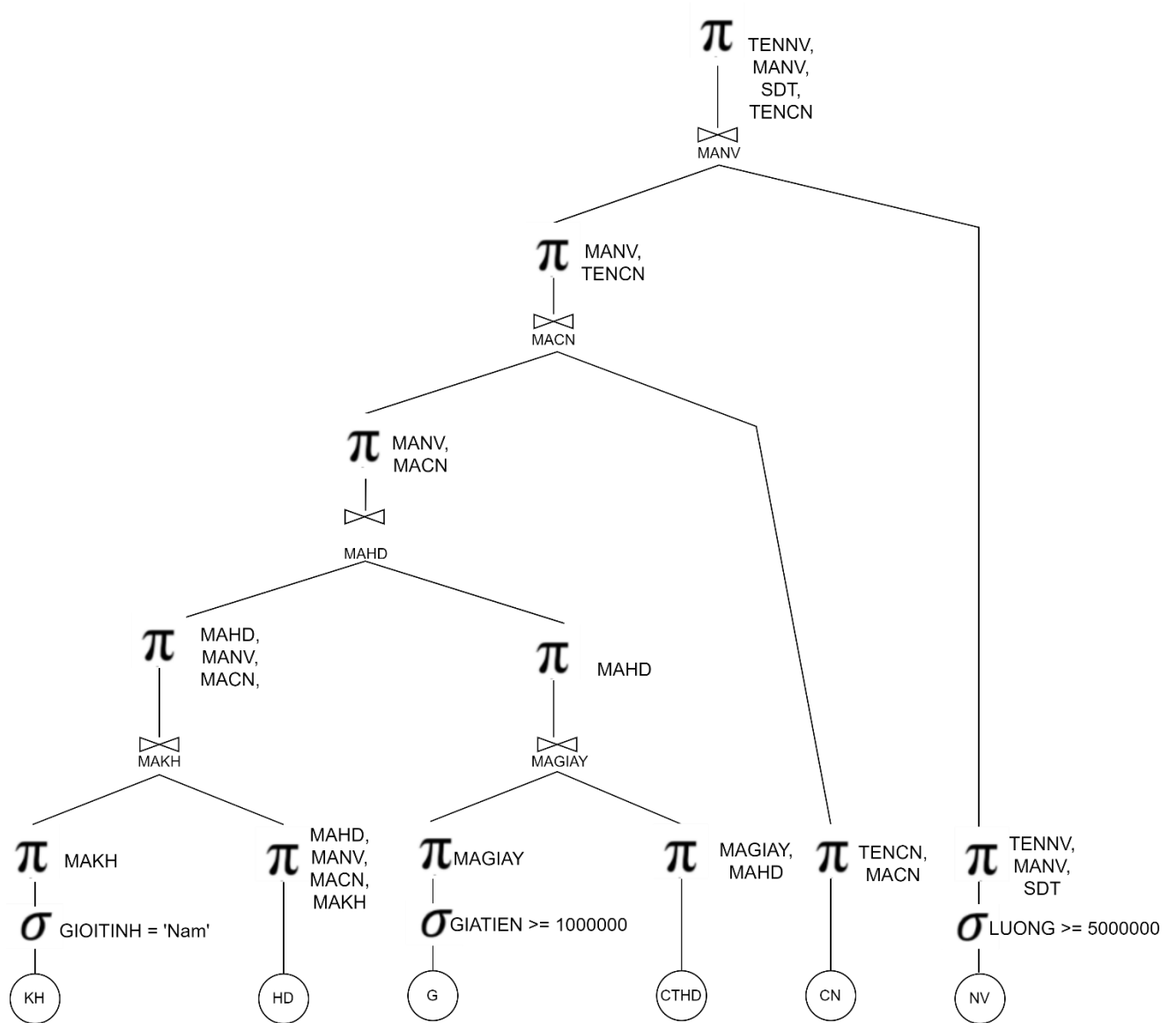
Đồ thị truy vấn liên thông nên câu truy vấn đúng ngữ nghĩa



## 4.2 Phân rã câu truy vấn ban đầu



### 4.3 Câu truy vấn đại số quan hệ sau khi tối ưu hóa toàn cục



CODE:

```
explain plan for
select nv.manv, nv.tennv, nv.sodt, tcn.tencn
from(
  select cn.tencn, nhd.manv
  from(
    select
    mhd.macn, mhd.manv
    from(
      select hd.mahd, hd.manv, hd.makh, hd.macn
      from(
        select mahd, manv, makh, macn
        from hoadon
      ) hd join (
        select makh
        from khachhang
        where gioitinh = 'Nam'
      ) kh ON hd.makh = kh.makh
    ) mhd join(
      select cthd.mahd, cthd.magiay
      from(
        select mahd, magiay
        from cthd
      ) cthd join (
        select magiay
        from giay
        where giatien >= 1000000
      ) g ON cthd.magiay = g.magiay
    ) ct ON mhd.mahd = ct.mahd
  ) nhd join (
    select macn, tencn
    from chinhanh
  ) cn on nhd.macn = cn.macn
) tcn join (
  select manv, tennv, sodt
  from nhanvien
  where luong > 5000000
) nv on tcn.manv = nv.manv;
```

## 4.4 Đánh giá thông qua EXPLAIN plan

Chưa tối ưu hóa

	0		SELECT STATEMENT									
*	1		HASH JOIN				15		995K	995K	789K	(0)
*	2		HASH JOIN				15		1209K	1209K	1122K	(0)
*	3		HASH JOIN				15		1281K	1281K	1178K	(0)
*	4		HASH JOIN				16		1222K	1222K	1445K	(0)
	5		MERGE JOIN				22					
	6		TABLE ACCESS BY INDEX ROWID		CHINHANH		1					
	7		INDEX FULL SCAN		PK_CN		1					
*	8		SORT JOIN				22		2048		2048	(0)
	9		TABLE ACCESS FULL		HOADON		22					
*	10		TABLE ACCESS FULL		KHACHHANG		8					
	11		INDEX FULL SCAN		PK_CTHD		20					
*	12		TABLE ACCESS FULL		NHANVIEN		15					
*	13		TABLE ACCESS FULL		GIAY		25					

2 Predicate Information (identified by operation id):

3	-----
4	
5	1 - access("MAGIAY"="MAGIAY")
6	2 - access("MANV"="MANV")
7	3 - access("MAHD"="MAHD")
8	4 - access("MAKH"="MAKH")
9	8 - access("MACN"="MACN")
10	filter("MACN"="MACN")
11	10 - filter("GIOITINH"='Nam')
12	12 - filter("LUONG">5000000)
13	13 - filter("GIATIEN">=1000000)

Sau khi tối ưu

PLAN_TABLE_OUTPUT									
24		0		SELECT STATEMENT					
25	*	1		HASH JOIN				15	995K  995K  768K (0)
26	*	2		HASH JOIN				15	1209K  1209K  855K (0)
27	*	3		HASH JOIN				15	1281K  1281K  899K (0)
28	*	4		HASH JOIN				16	1222K  1222K  1092K (0)
29		5		MERGE JOIN				22	
30		6		TABLE ACCESS BY INDEX ROWID		CHINHANH		1	
31		7		INDEX FULL SCAN		PK_CN		1	
32	*	8		SORT JOIN				22	2048   2048   2048 (0)
33		9		TABLE ACCESS FULL		HOADON		22	
34	*	10		TABLE ACCESS FULL		KHACHHANG		8	
35		11		INDEX FULL SCAN		PK_CTHD		20	
36	*	12		TABLE ACCESS FULL		NHANVIEN		15	
37	*	13		TABLE ACCESS FULL		GIAY		25	
38	-----								
Predicate Information (identified by operation id):									
-----									
1 - access("G"."MAGIAY"="CTHD"."MAGIAY")									
2 - access("NV"."MANV"="HD"."MANV")									
3 - access("HD"."MAHD"="CTHD"."MAHD")									
4 - access("KH"."MAKH"="HD"."MAKH")									
8 - access("CN"."MACN"="HD"."MACN")									
filter("CN"."MACN"="HD"."MACN")									
10 - filter("KH"."GIOITINH"='Nam')									
12 - filter("NV"."LUONG">5000000)									
13 - filter("G"."GIATIEN">=1000000)									

Nhận xét:

#### 4.5 Lược đồ phân mảnh

Quan hệ CHINHANH phân mảnh ngang chính theo tên chi nhánh:

$$CN1 = \sigma_{TenCN = 'Quan\ 9,\ TPHCM'} CHINHANH$$

$$CN2 = \sigma_{TenCN = 'Quan\ 1,\ TPHCM'} CHINHANH$$

Quan hệ G\_QLKHO, G\_QLBH, HOADON, MAHOADON, KHACHHANG, NHANVIEN phân mảnh ngang dẫn xuất như sau:

$$QLK1 = G\_QLKHO \bowtie_{MACN\ CN1}$$

$$QLK2 = G\_QLKHO \bowtie_{MACN\ CN2}$$

$$QLBH1 = G\_QLBH \bowtie_{MACN\ CN1}$$

$$QLBH2 = G\_QLBH \bowtie_{MACN\ CN2}$$

$$HD1 = HOADON \bowtie_{MACN\ CN1}$$

$$HD2 = HOADON \bowtie_{MACN\ CN2}$$

$$CT1 = HD1 \bowtie_{MAHD\ CN1}$$

$$CT2 = HD2 \bowtie_{MAHD\ CN2}$$

$$KH1 = KHACHHANG \bowtie_{MACN\ CN1}$$

$$KH2 = KHACHHANG \bowtie_{MACN\ CN2}$$

$$NV1 = NHANVIEN \bowtie_{MACN\ CN1}$$

$$NV2 = NHANVIEN \bowtie_{MACN\ CN2}$$

Quan hệ KHACHHANG phân mảnh hỗn hợp như sau:

$$KH1A = \pi_{TENKH, DIACHI, SODT} (KHACHHANG \bowtie_{MACN\ CN1})$$

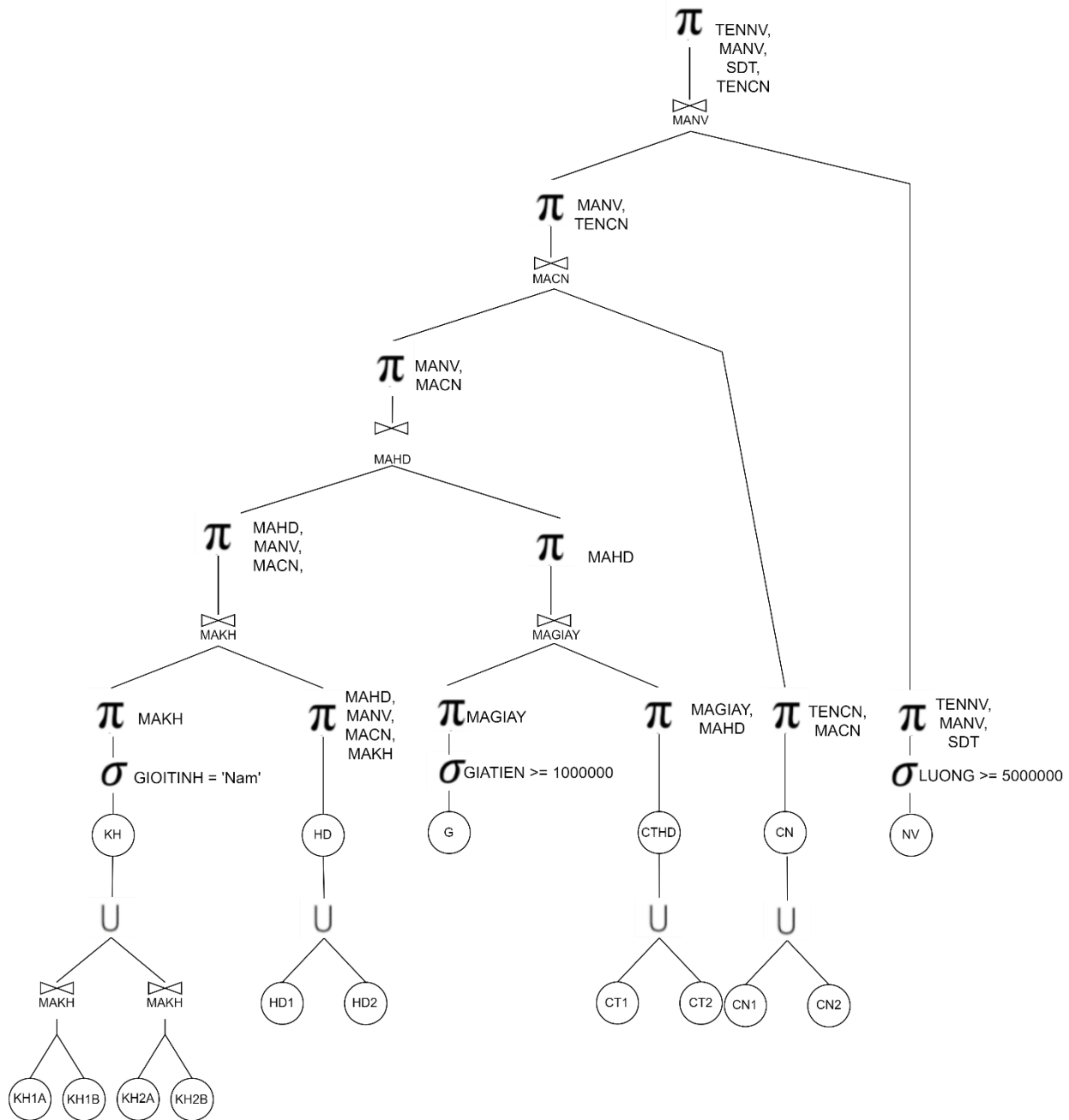
$$KH1B = \pi_{MAKH, TUOI, GIOITINH} (KHACHHANG \bowtie_{MACN\ CN1})$$

$$KH2A = \pi_{TENKH, DIACHI, SODT} (KHACHHANG \bowtie_{MACN\ CN2})$$

$$KH2B = \pi_{MAKH, TUOI, GIOITINH} (KHACHHANG \bowtie_{MACN\ CN2})$$

Quan hệ GIAY, NHANVIEN được nhân bản tại tất cả chi nhánh

#### 4.6 Cây truy vấn dựa vào lược đồ phân mảnh



## CHƯƠNG 5: CƠ CHẾ NHÂN BẢN TRONG ORACLE

### 5.1 Khái niệm

Nhân bản cơ sở dữ liệu Oracle không là gì ngoài một quy trình được sử dụng để tạo cơ sở dữ liệu giống hệt cơ sở dữ liệu Oracle hiện có. Kỹ thuật nhân bản được sử dụng để tạo một bản sao của Cơ sở dữ liệu hiện có để thực hiện các hoạt động thử nghiệm khác nhau của DBA như sao lưu và phục hồi.

Cách thức hoạt động:

Phương pháp nhân bản	Cách thức hoạt động
Recovery Manager (RMAN) Backup	<ul style="list-style-type: none"><li>- Kết nối các phiên bản Oracle nguồn và đích</li><li>- Sao chép các tệp cơ sở dữ liệu bằng tính năng sao chép RMAN</li><li>- Phục hồi và mở cơ sở dữ liệu nhân bản</li></ul>
Staging Areas	<ul style="list-style-type: none"><li>- Sao lưu từng tệp cơ sở dữ liệu và lưu trữ nó trong khu vực tổ chức</li><li>- Chuyển từng tệp sao lưu từ nguồn đến đích</li><li>- Khôi phục từng tệp sao lưu vào các vị trí đã chỉ định</li><li>- Phục hồi và mở cơ sở dữ liệu nhân bản</li></ul>
An Existing Backup	<ul style="list-style-type: none"><li>- Tạo cơ sở dữ liệu nhân bản tại thời điểm cụ thể hoặc SCN</li><li>- Xác thực các bản sao lưu trước khi thực hiện thao tác sao chép</li><li>- Chuyển các tệp nhật ký làm lại được lưu trữ cần thiết sang máy chủ đích</li><li>- Phục hồi và mở cơ sở dữ liệu nhân bản</li></ul>

### 5.2 Cơ chế nhân bản của RMAN

#### 5.2.1 Cơ chế nhân bản của Recovery Manager (RMAN) Backup

- Tạo một control file cho cơ sở dữ liệu nhân bản
- Khôi phục các data file đích vào cơ sở dữ liệu nhân bản
- Thực hiện phục hồi không đầy đủ bằng cách sử dụng tất cả các bản sao lưu incremental có sẵn và các archived redo log files
- Tắt và khởi động lại instance phụ trợ
- Mở cơ sở dữ liệu nhân bản với tùy chọn RESETLOGS
- Tạo các online redo log files
- Tạo một định danh cơ sở dữ liệu mới, duy nhất (DBID) cho cơ sở dữ liệu nhân bản

#### 5.2.2 Các bước tiến hành

- Tạo tệp mật khẩu trên máy đích
- Thiết lập kết nối giữa máy chủ và máy đích (tnsnames.ora, sqlnet.ora)
- Tạo các thư mục cho các tệp cơ sở dữ liệu
- Lấy bản sao lưu RMAN từ máy chủ và sao chép nó vào máy đích.
- Khởi động cơ sở dữ liệu đích ở chế độ Nomount
- Chạy lệnh nhân bản cơ sở dữ liệu RMAN

Link video tham khảo:

[https://vimeo.com/12847611?embedded=true&source=video\\_title&owner=3403610](https://vimeo.com/12847611?embedded=true&source=video_title&owner=3403610)



## TÀI LIỆU THAM KHẢO

1. Slide bài giảng và hướng dẫn thực hành do giảng viên cung cấp.

2. Tài liệu tham khảo giáo viên cung cấp.

3. Tài liệu tạo hàm:

[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_5009.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5009.htm)

4. Tài liệu tạo thủ tục:

[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_6009.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6009.htm)

5. Tài liệu về nhân bản cơ sở dữ liệu:

<https://docs.oracle.com/en/enterprise-manager/cloud-control/enterprise-manager-cloud-control/13.3.1/emlcm/cloning-oracle-databases-and-pluggable-databases.html#GUID-DDDC435F-C89B-482C-91CC-34C03C53099D>

<https://www.youtube.com/watch?v=g6aVyKYoIsI&t=554s>