ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

CƠ SỞ DỮ LIỆU PHÂN TÁN
BÀI TẬP THỰC HÀNH LỚN 1

Lớp: IS211.M11

GVHD: Nguyễn Minh Nhựt

Nhóm: 3

Sinh viên thực hiện:

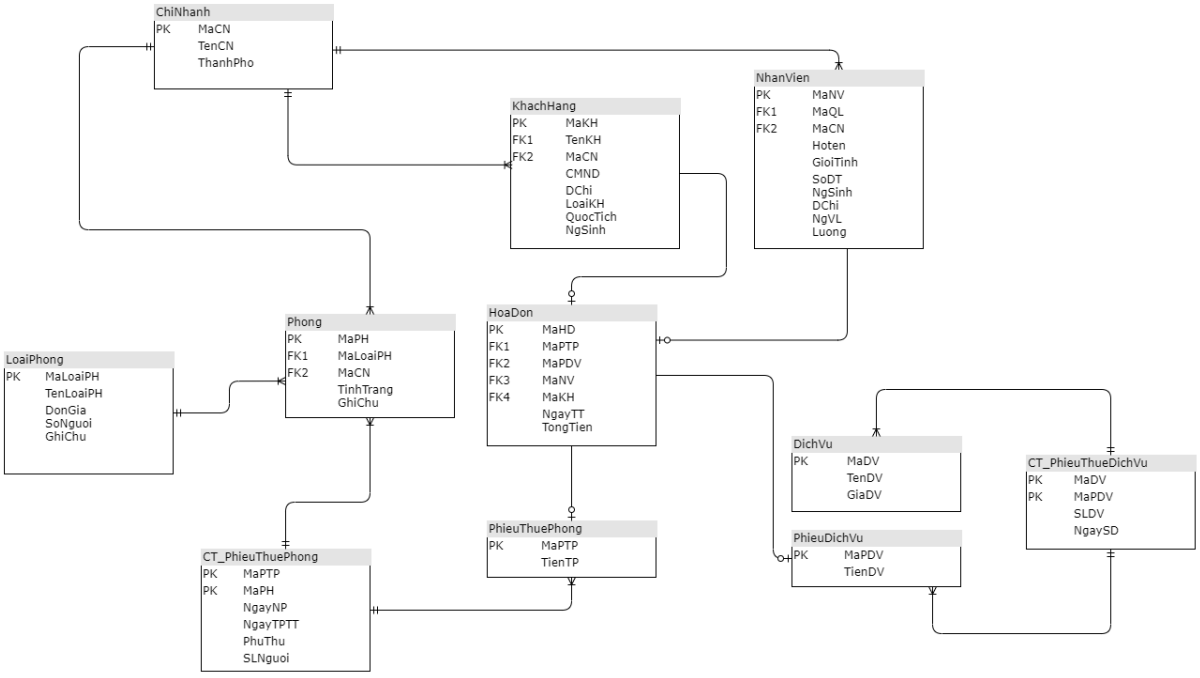
- Đặng Vũ Phương Uyên – 19520345
- Lê Thị Ái Nhi – 19521963
- Trần Kim Ngân – 19521890
- Nguyễn Thị Thu Phương – 19522066

TP. Hồ Chí Minh, Ngày 09 tháng 12 năm 2021

MỤC LỤC

I. Thiết kế CSDL Phân tán và truy vấn trên môi trường máy ảo, radmin

1. Mô hình vật lý.



2. Mô hình dữ liệu quan hệ .

DichVu (**MaDV**, tenDV, GiaDV);
HoaDon (**MaHD**, TongTien, NgayTT, MaNV, MaKH, MaPTP, MaPDV);
KhachHang (**MaKH**, TenKH, CMND, QuocTich, NgSinh, DChi, LoaiKH, MaCN);
NhanVien (**MaNV**, HoTen, GioiTinh, SoDT, NgSinh, DChi, NgVL, Luong, MaQL, MaCN);
PhieuDichVu (**MaPDV**, TienDV);
PhieuThuePhong (**MaPTP**, TienTP);
Phong (**MaPH**, TinhTrang, GhiChu, MaLoaiPH, MaCN);
CT_PhieuThueDV (**MaPDV, MaDV**, SLDV, NgaySD);
CT_PhieuThuePhong **(MaPTP, MaPH**, NgayNP, NgayTPTT, PhuThu, SLNguoi);

3. Tạo và gán quyền cho hai chi nhánh.

CHI NHÁNH 2	CHI NHÁNH 3
CREATE USER C##CN2 IDENTIFIED BY CN2; GRANT CONNECT,RESOURCE TO C##CN2; GRANT DBA TO C##CN2; GRANT CREATE PUBLIC DATABASE LINK TO C##CN2;	CREATE USER C##CN3 IDENTIFIED BY CN3; GRANT CONNECT,RESOURCE TO C##CN3; GRANT DBA TO C##CN2; GRANT CREATE PUBLIC DATABASE LINK TO C##CN3;

4. Các câu truy vấn

- Số dòng dữ liệu 20 dòng dữ liệu trở lên với mỗi bảng.
- Thực hiện 10 câu truy vấn nhiều dạng (PHẢI có HỘI, GIAO, TRỪ, CHIA, HÀM GOM NHÓM VÀ TÍNH TOÁN)

a. Thống kê doanh thu của từng tuần từ đầu tháng đến ngày hiện tại (phép hội)

- Ý nghĩa : Để theo dõi doanh thu của khách sạn của tháng đó nhằm đánh giá tình hình kinh doanh và tiến hành các điều chỉnh nếu có.

```
SELECT *
FROM (
    SELECT SUM(TONGTIEN) DOANHTHU, WEEK, 'CN0002' AS CHINHANH
    FROM HOADON HD1 JOIN (   SELECT X, TRUNC( (X-NEXT_DAY(TRUNC(X,'MM')-
8, 'SUN'))/7 )+1 WEEK
                             FROM (  SELECT TRUNC(SYSDATE,'MM') +ROWNUM-1
X
                                     FROM ALL_OBJECTS
                                     WHERE ROWNUM <= (   SELECT
EXTRACT(DAY FROM LAST_DAY(SYSDATE)) AS LAST_DAY_OF_MNTH
                                                         FROM DUAL
                                                     )
                                 )
                     ) A ON EXTRACT(MONTH FROM A.X) = EXTRACT(MONTH FROM
HD1.NGAYTT)
    GROUP BY A.WEEK
    UNION
```

```
    SELECT SUM(TONGTIEN) DOANHTHU, WEEK, 'CN0003' AS CHINHANH
    FROM C##CN03.HOADON@BTL1 HD2 JOIN (   SELECT X, TRUNC( (X-
NEXT_DAY(TRUNC(X,'MM')-8, 'SUN'))/7 )+1 WEEK
                              FROM (  SELECT TRUNC(SYSDATE,'MM') +ROWNUM-1
X
                                      FROM ALL_OBJECTS
                                      WHERE ROWNUM <= (   SELECT
EXTRACT(DAY FROM LAST_DAY(SYSDATE)) AS LAST_DAY_OF_MNTH
                                                          FROM DUAL
                                                      )
                                  )
                      ) A ON EXTRACT(MONTH FROM A.X) = EXTRACT(MONTH FROM
HD2.NGAYTT)
    GROUP BY A.WEEK
)
ORDER BY CHINHANH, WEEK;
```

b. Thống kê số lượng phòng còn trống theo từng loại của từng chi nhánh (hàm gom nhóm)

- Ý nghĩa : Theo dõi các loại phòng còn phòng trống ở từng chi nhánh để giới thiệu cho khách hàng khi khách hàng đến thuê phòng hoặc đánh giá doanh thu của tháng để có kế hoạch chiến lược phù hợp thúc đẩy doanh thu.

```
SELECT CN.MACN, TENCN, TENLOAIPH, COUNT(P.MAPH)
FROM C##CN02.CHINHANH@BTL1 CN JOIN C##CN02.PHONG@BTL1 P ON CN.MACN =
P.MACN
        JOIN C##CN02.LOAIPHONG@BTL1 LP ON LP.MALOAIPH = P.MALOAIPH
WHERE TINHTRANG = 'Trống'
GROUP BY CN.MACN, TENCN, TENLOAIPH
UNION
SELECT CN3.MACN, TENCN, TENLOAIPH, COUNT(P3.MAPH)
FROM C##CN03.CHINHANH CN3 JOIN C##CN03.PHONG P3 ON CN3.MACN = P3.MACN
```

```
        JOIN C##CN03.LOAIPHONG LP3 ON LP3.MALOAIPH = P3.MALOAIPH

WHERE TINHTRANG = 'Trống'

GROUP BY CN3.MACN, TENCN, TENLOAIPH
```

c. Hiển thị danh sách nhân viên ưu tú nhất của chi nhánh 3 trong tháng hiện tại (nhân viên thực hiện lập hóa đơn nhiều nhất)(hàm gom nhóm)

- Ý nghĩa : Tìm ra những nhân viên xuất sắc để tuyên dương và khen thưởng.

```
SELECT  EXTRACT(Month FROM NGAYTT) as thang,CN2.MACN, CN2.TENCN,
NV2.MANV, NV2.HoTen, COUNT(HD2.MAHD) AS SL_HOADON

        FROM C##CN03.HOADON@BTL1 HD2, C##CN03.NHANVIEN@BTL1 NV2,
C##CN03.CHINHANH@BTL1 CN2

        WHERE HD2.MANV = NV2.MANV AND CN2.MACN = NV2.MACN AND
EXTRACT(Month FROM SYSDATE) = EXTRACT(Month FROM NGAYTT)

        GROUP BY  EXTRACT(Month FROM HD2.NGAYTT), CN2.MACN, CN2.TENCN,
NV2.MANV,NV2.HOTEN

        HAVING COUNT (HD2.MAHD) >= ALL(  SELECT COUNT (HD3.MAHD)

                                         FROM C##CN03.HOADON@BTL1 HD3

                                         WHERE  EXTRACT(Month FROM
HD3.NGAYTT)= EXTRACT(Month FROM SYSDATE)

                                         GROUP BY HD3.MANV

                                    );
```

d. Tìm khách hàng đã từng thuê tất cả các loại phòng ở chi nhánh 1 và chi nhánh 2 nhưng chưa từng thuê phòng ở chi nhánh 3.(phép chia và phép trừ, phép giao)

- Ý nghĩa : Tìm ra khách hàng đó để áp dụng khuyến mãi cho họ khi họ đến thuê phòng ở chi nhánh 3.

```
SELECT *

FROM C##CN01.KHACHHANG_HIENTHI@BTL KH

WHERE NOT EXISTS (

                        SELECT LP.MALOAIPH

                        FROM C##CN01.LOAIPHONG@BTL LP JOIN
C##CN01.PHONG@BTL P ON LP.MALOAIPH = P.MALOAIPH
```

```
                                JOIN CT_PHIEUTHUEPHONG CTPTP ON P.MAPH =
CTPTP.MAPH

                        WHERE  NOT EXISTS (

                                                        SELECT *

                                                        FROM
C##CN01.HOADON@BTL HD

                                                        WHERE HD.MAKH =
KH.MAKH AND HD.MAPTP = CTPTP.MAPTP

                                                        )

                        )

INTERSECT

(SELECT *

FROM C##CN02.KHACHHANG_HIENTHI@BTL1 KH1

WHERE NOT EXISTS (

                        SELECT LP1.MALOAIPH

                        FROM C##CN02.LOAIPHONG@BTL1 LP1 JOIN
C##CN02.PHONG@BTL1 P1 ON LP1.MALOAIPH = P1.MALOAIPH

                                JOIN CT_PHIEUTHUEPHONG CTPTP1 ON P1.MAPH
= CTPTP1.MAPH

                        WHERE  NOT EXISTS (

                                                        SELECT *

                                                        FROM
C##CN02.HOADON@BTL1 HD1

                                                        WHERE HD1.MAKH =
KH1.MAKH AND HD1.MAPTP = CTPTP1.MAPTP

                                                        )

                        )
```

```
)

MINUS

SELECT *

FROM KHACHHANG_HIENTHI

WHERE MAKH IN (SELECT MAKH

                FROM HOADON HD2

                WHERE MAPTP IS NOT NULL

                GROUP BY MAKH

                HAVING  COUNT(*) > 0

                );
```

e. Tìm kiếm những khách hàng VIP (khách hàng đã và đang sử dụng dịch vụ trong vòng 1 năm) của khách sạn. (phép hội)

- Ý nghĩa : Có tác dụng thống kê, phân loại các loại khách hàng để gửi các thông báo chương trình khuyến mãi phù hợp với từng loại khách hàng.

```
SELECT *

FROM (

    SELECT KH1.MAKH, KH1.TENKH, KH1.LOAIKH, CASE WHEN KH1.LOAIKH =
'thanhvien' THEN 'Uu dai cua thanh vien'

                                                ELSE 'Uu dai cua khach
hang'

                                                END

    FROM KHACHHANG_LUUTRU KH1 JOIN ( SELECT MAKH, MAX(NGAYTT) AS
NGAYSUDUNG

                                    FROM HOADON

                                    GROUP BY MAKH

                                    ) HD1 ON HD1.MAKH = KH1.MAKH

    WHERE (SYSDATE - 365) <= NGAYSUDUNG AND NGAYSUDUNG <= SYSDATE

    UNION
```

```
    SELECT KH2.MAKH, KH2.TENKH, KH2.LOAIKH, CASE WHEN KH2.LOAIKH =
'thanhvien' THEN 'Uu dai cua thanh vien'
                                                    ELSE 'Uu dai cua khach
hang'
                                                    END
    FROM C##CN03.KHACHHANG_LUUTRU@BTL1 KH2 JOIN (SELECT MAKH, MAX(NGAYTT)
AS NGAYSUDUNG
                                        FROM C##CN03.HOADON@BTL1
                                        GROUP BY MAKH
                                        ) HD2 ON HD2.MAKH = KH2.MAKH
    WHERE (SYSDATE - 365) <= NGAYSUDUNG AND NGAYSUDUNG <= SYSDATE
    )
ORDER BY MAKH
```

f. Thống kê top 3 dịch vụ được sử dụng nhiều nhất trong khách sạn (tính toán và gom nhóm)

- Ý nghĩa : Tìm ra những dịch vụ được sử dụng nhiều nhất để tập trung đầu tư vào các loại hình dịch vụ đó, đồng thời nâng cao chất lượng dịch vụ của các dịch vụ ít được sử dụng hơn để thu hút khách hàng sử dụng.

```
SELECT MADV
FROM (
      SELECT A.MADV, SUM(SLDV)
      FROM (
            SELECT MADV, SLDV
            FROM CT_PHIEUDICHVU
            UNION
            SELECT MADV,SLDV
            FROM C##CN02.CT_PHIEUDICHVU@BTL1
            ) A
      GROUP BY MADV
      ORDER BY SUM(SLDV) DESC
      )
WHERE ROWNUM <= 3;
```

g. Tìm khách hàng có quốc tịch là Việt Nam thuê phòng với số lượng người trong phòng đó bằng với số người quy định của loại phòng đó từ ngày 01/09/2021 đến ngày 02/09/2021 và có sử dụng ít nhất 2 dịch vụ của khách sạn trong thời gian thuê (gom nhóm và phép hội)

- Ý nghĩa : Để áp dụng các khuyến mãi dịch vụ và thuê phòng người Việt Nam may mắn thuê phòng vào ngày 02/09 để mừng ngày Quốc Khánh.

```
SELECT MAKH,TENKH

FROM

    (SELECT KH.MAKH,KH.TENKH

    FROM HOADON HD, KHACHHANG_HIENTHI KH, PHIEUTHUEPHONG PTP,
PHIEUDICHVU PDV

    WHERE HD.MAKH = KH.MAKH AND PTP.MAPTP = HD.MAPTP AND PDV.MAPDV =
HD.MAPDV

        AND KH.QUOCTICH = 'Việt Nam' AND  PDV.MAPDV IN ( SELECT
CT_PDV.MAPDV

                                                        FROM CT_PHIEUDICHVU
CT_PDV

                                                        GROUP BY
CT_PDV.MAPDV

                                                        HAVING COUNT (MADV)
>= 2) AND PTP.MAPTP IN (


SELECT CT_PTP.MAPTP


FROM CT_PHIEUTHUEPHONG CT_PTP,PHONG P,LOAIPHONG LP


WHERE CT_PTP.MAPH = P.MAPH AND P.MALOAIPH= LP.MALOAIPH AND CT_PTP.SLNGUOI
= LP.SONGUOI


AND CT_PTP.NGAYNP BETWEEN '15-FEB-2021' AND '26-FEB-2021'


)

UNION

SELECT KH2.MAKH,KH2.TENKH

FROM C##CN03.HOADON@BTL1 HD2, C##CN03.KHACHHANG_HIENTHI@BTL1 KH2,
C##CN03.PHIEUTHUEPHONG@BTL1 PTP2, C##CN03.PHIEUDICHVU@BTL1 PDV2
```

```
WHERE HD2.MAKH = KH2.MAKH AND PTP2.MAPTP = HD2.MAPTP AND PDV2.MAPDV =
HD2.MAPDV

AND KH2.QUOCTICH = 'Việt Nam' AND  PDV2.MAPDV IN ( SELECT CT_PDV2.MAPDV
                                                    FROM
C##CN03.CT_PHIEUDICHVU@BTL1 CT_PDV2
                                                    GROUP BY
CT_PDV2.MAPDV
                                                    HAVING COUNT
(MADV) >= 2) AND PTP2.MAPTP IN (


SELECT CT_PTP2.MAPTP


FROM C##CN03.CT_PHIEUTHUEPHONG@BTL1 CT_PTP2, C##CN03.PHONG@BTL1 P2,
C##CN03.LOAIPHONG@BTL1 LP2


WHERE CT_PTP2.MAPH = P2.MAPH AND P2.MALOAIPH= LP2.MALOAIPH AND
CT_PTP2.SLNGUOI = LP2.SONGUOI


AND CT_PTP2.NGAYNP BETWEEN '15-FEB-2021' AND '26-FEB-2021'


GROUP BY CT_PTP2.MAPTP))

GROUP BY MAKH, TENKH;
```

h. Tìm tên nhân viên, số hóa đơn, tổng tiền đã làm việc ít nhất 2 năm và lập hóa đơn trên 30 triệu của khách hàng có quốc tịch khác Việt Nam và là lần đầu tiên sử dụng dịch vụ ở khách sạn.(gom nhóm và phép hội)

- Ý nghĩa : Tìm ra những nhân viên thường lập hóa đơn có giá trị cao cho khách hàng người nước ngoài lần đầu tiên sử dụng dịch vụ tại khách sạn để tìm hiểu thông tin từ nhân viên về nhóm khách hàng này.

```
SELECT NV.HOTEN, MAHD, TONGTIEN

FROM C##CN02.NHANVIEN@BTL1 NV JOIN (

                        SELECT HD1.MAHD, HD1.TONGTIEN, HD1.MANV

                        FROM C##CN02.HOADON@BTL1 HD1 JOIN (
```

```
                                                    SELECT KH.MAKH,
COUNT(MAHD)

                                                    FROM
C##CN02.HOADON@BTL1 HD JOIN C##CN02.KHACHHANG_HIENTHI@BTL1 KH ON
HD.MAKH = KH.MAKH

                                                    WHERE QUOCTICH <> 'Việt
Nam'

                                                    GROUP BY KH.MAKH

                                                    HAVING COUNT(MAHD) = 1

                                                ) A ON HD1.MAKH = A.MAKH

                            ) B ON B.MANV = NV.MANV

UNION

SELECT NV2.HOTEN, MAHD, TONGTIEN

FROM C##CN03.NHANVIEN NV2 JOIN (

                            SELECT HD2.MAHD, HD2.TONGTIEN, HD2.MANV

                            FROM C##CN03.HOADON HD2 JOIN (

                                                    SELECT KH2.MAKH,
COUNT(MAHD)

                                                    FROM C##CN03.HOADON HD3
JOIN C##CN03.KHACHHANG_HIENTHI KH2 ON HD3.MAKH = KH2.MAKH

                                                    WHERE QUOCTICH <> 'Việt
Nam'

                                                    GROUP BY KH2.MAKH

                                                    HAVING COUNT(MAHD) = 1

                                                ) A2 ON HD2.MAKH = A2.MAKH

                            ) B2 ON B2.MANV = NV2.MANV
```

i. Tìm hóa đơn được lập vào ngày 1/6/2021 *có mà có tính phí phụ thu của các phòng có đơn giá trên 1tr500 do nhân viên có số điện thoại là 0369275023 lập*

- Ý nghĩa : Vào ngày 1/6/2021 , khách sạn không thu phí phụ thu cho các phòng có đơn giá trên 1tr500 nhưng do nhầm lẫn , nhân viên có số điện thoại '0369275023' đã thu phí phụ thu này nên tiến hành tìm các hóa đơn do nhân viên này lập ra vào ngày đó để hoàn trả lại phí phụ thu lại cho khách hàng.

```
SELECT HD2.MAHD

FROM C##CN03.HOADON@BTL1 HD2, C##CN03.NHANVIEN@BTL1 NV2,
C##CN03.PHIEUTHUEPHONG@BTL1 PTP2, C##CN03.CT_PHIEUTHUEPHONG@BTL1 CT_PTP2

WHERE HD2.MANV = NV2.MANV AND PTP2.MAPTP = HD2.MAPTP AND PTP2.MAPTP =
CT_PTP2.MAPTP

AND EXTRACT (MONTH FROM HD2.NGAYTT ) = 1  AND NV2.SODT= '0969482744' AND
CT_PTP2.PHUTHU IN (SELECT PHUTHU


FROM C##CN03.CT_PHIEUTHUEPHONG@BTL1 CT_PTP3


WHERE   PTP2.MAPTP = CT_PTP3.MAPTP  AND PHUTHU > 0)

GROUP BY HD2.MAHD;
```

j. Tìm phiếu dịch vụ có tổng tiền chiếm từ 30 - 40 % của hóa đơn khách hàng lập ngày '1/12/2021' mà khách hàng đó có quốc tịch là Việt Nam. (tính toán)

- Ý nghĩa : Tìm những khách hàng chi nhiều cho dịch vụ của khách sạn vào ngày '01/12/2021' để tặng phiếu khuyến mãi cho khách hàng đó.

```
SELECT A.MAPDV, A.TIENDV, B.TONGTIEN

FROM PHIEUDICHVU A, HOADON B, KHACHHANG_HIENTHI C

WHERE A.MAPDV= B.MAPDV AND B.MAKH = C.MAKH AND

A.TIENDV >= B.TONGTIEN*30/100 AND A.TIENDV <= B.TONGTIEN*40/100

AND TO_DATE(B.NGAYTT)='01-DEC-2021' AND C.QUOCTICH ='Việt Nam';
```

II. Viết hàm, thủ tục, ràng buộc toàn vẹn truy vấn trên môi trường phân tán

- ❖ Viết 01 hàm, 01 thủ tục trên môi trường phân tán
- ❖ Viết 01 RBTV (Trình bày trong báo cáo Bối cảnh, nội dung, bảng tầm hưởng)

1. Store Procedure: *Hiển thị thông tin phòng còn trống của các chi nhánh dựa theo giá trị nhập vào.*

- Tên: SHOW_INFORMATION_VACANCIES.
- Nội dung: Liệt kê các thông tin phòng còn trống tương ứng với chi nhánh nhập vào.
- Các bước thực hiện:

[1]: Gán giá trị cho biến THONG_TIN_CN_VAL với giá trị bằng 0, NOT_EMPTY với giá trị bằng 0.

[2]: Tiến hành thực hiện thủ tục THONG_TIN_CN.

[2.1]: Nếu giá trị mã chi nhánh nhập vào là null, thì sẽ lấy tất cả thông tin bao gồm mã chi nhánh và tên chi nhánh theo phân mảnh ngang của hệ thống khách sạn.

[2.1.1]: Gán kết quả của câu lệnh truy vấn thông tin tất các chi nhánh vào biến V_CHINHANH.

[2.1.2]: Thay đổi giá trị của biến THONG_TIN_CN_VAL = 1.

[2.1.3]: Nhảy sang bước [3].

[2.2]: Nếu giá trị mã chi nhánh nhập và khác null thì tiến hành kiểm tra chi nhánh có mã phù hợp.

[2.2.1]: Đếm các bản ghi trong quan hệ ChiNhanh của phân mảnh ngang ở Hà Nội với điều kiện là mã chi nhánh của chi nhánh thứ nhất bằng với giá trị mã chi nhánh nhập vào và gán vào biến NOT_EMTPY.

[2.2.1.1]: Nếu biến NOT_EMPTY bằng 1, tức là có tồn tại mã chi nhánh đó trong phân vùng của khu vực Hà Nội.

[2.2.1.1.1]: Gán kết quả của câu lệnh truy vấn thông tin của chi nhánh 1 vào biến V_CHINHANH và thay đổi giá trị của biến THONG_TIN_CN_VAL.

[2.2.1.1.2]: Nhảy sang bước [3].

[2.2.1.2]: Ngược lại, tiếp tục đếm các bản ghi trong quan hệ ChiNhanh của phân mảnh ngang ở Huế với điều kiện là mã chi nhánh của chi nhánh thứ hai bằng với giá trị mã chi nhánh nhập vào và gán vào biến NOT_EMTPY.

[2.2.1.2.1]: Nếu biến NOT_EMPTY bằng 1, tức là có tồn tại mã chi nhánh đó trong phân vùng của khu vực Hà Nội.

[2.2.1.2.1]: Gán kết quả của câu lệnh truy vấn thông tin của chi nhánh 1 vào biến V_CHINHANH và thay đổi giá trị của biến THONG_TIN_CN_VAL.

[2.2.1.2.2]: Nhảy sang bước [3].

[3]: Kiểm tra giá trị của biến THONG_TIN_CN_VAL.

[3.1]: Nếu giá trị của biến bằng 0 thì xuất thông báo “Không tồn tại chi nhánh cần tìm”.

[3.2]: Ngược lại, tiên hành thực hiện thủ tục SHOW_INFORMATION_ROOM. Và nhảy sang bước [4].

[4]: Kiểm tra mã tình trạng phòng sẽ được nhập vào:

[4.1]: Thực hiện xuất thông tin chi nhánh và các phòng trống tương ứng. Đầu tiên, tiến hành kiểm tra giá trị nhập từ bàn phím.

[4.1.1]: Nếu giá trị của biến đầu vào là null thì tiến hành thực hiện thủ tục HIEN_THI_PHONG_BA_CN. Và ngảy sang bước [5].

[4.2.1]: Ngược lại, chạy hàm HIEN_THI_PHONG_MOT_CN và nhảy sang bước [6].

[5]: Trong thủ tục HIEN_THI_PHONG_BA_CN dùng dể gán các kết quả phòng trống vào biển V_PHONG_CHINHANH và đồng thời đếm các giá trị trong biến V_PHONG_CHINHANH đó rồi chuyển sang bước [7].

[6]: Trong hàm HIEN_THI_PHONG_MOT_CN sẽ gán giá trị phòng trống cảu một chi nhánh cu thể vào biến V_PHONG_CHINHANH. Tuy nhiên, tại đây, khi có giá trị phòng trống của chi nhánh phù hợp thì biến NOT_EMPTY có giá trị khác không, ngược lại biến NOT_EMPTY không thay đổi và chuyển sang bước [7].

[7]: Hiển thị các giá trị phòng còn trống với điều kiện mã chi nhanh của phòng đó bằng với mã chi nhánh hiện tại của vòng lặp.

[8]: Tiến hành kiểm tra giá trị của biến NOT_EMPTY.

[8.1]: Nếu biến NOT_EMPTY bằng 0 thì xuất ra thông báo ‘Không tồn tại dữ liệu phòng còn trống’ và chuyển sang bước [9].

[8.2]: Ngược lại thì nhảy sang bước[9].

[9]: Kết thúc thủ tục.

```
create or replace PROCEDURE SHOW_INFORMATION_VACANCIES (V_MACN
CHINHANH.MACN%TYPE)
IS
    NOT_EMPTY INT;
    TYPE T_CHINHANH IS TABLE OF CHINHANH%ROWTYPE INDEX BY
PLS_INTEGER;
    TYPE T_KETHOP_PHONG_LOAIPHONG IS RECORD ( MAPH
PHONG.MAPH%TYPE,
                    TENLOAIPH LOAIPHONG.TENLOAIPH%TYPE,
                    DONGIA LOAIPHONG.DONGIA%TYPE,
                    SONGUOI LOAIPHONG.SONGUOI%TYPE,
                    MACN CHINHANH.MACN%TYPE
                                            );
    TYPE T_PHONG IS TABLE OF T_KETHOP_PHONG_LOAIPHONG INDEX BY
PLS_INTEGER;
    V_CHINHANH T_CHINHANH;
    V_PHONG_CHINHANH T_PHONG;
    THONG_TIN_CN_VAL INT;

    PROCEDURE THONG_TIN_CN IS
        BEGIN
            IF V_MACN IS NULL THEN
```

```
                BEGIN
                        SELECT MACN, TENCN, THANHPHO BULK COLLECT
INTO V_CHINHANH
                        FROM (
                                SELECT MACN, TENCN, THANHPHO
                                FROM
C##PUN_CN1.CHINHANH@DB_LINK_PUN_CN1
                                UNION
                                SELECT MACN, TENCN, THANHPHO
                                FROM C##PUN_CN2.CHINHANH
                                )
                        ORDER BY MACN;
                        THONG_TIN_CN_VAL:=1;
                    END;
                END IF;
            IF V_MACN IS NOT NULL THEN
                BEGIN
                    SELECT COUNT(*) INTO NOT_EMPTY
                    FROM C##PUN_CN1.CHINHANH@DB_LINK_PUN_CN1
                    WHERE MACN = V_MACN;
                    IF NOT_EMPTY = 1 THEN
                        BEGIN
                            SELECT  MACN,  TENCN,  THANHPHO  BULK
COLLECT INTO V_CHINHANH
                            FROM
C##PUN_CN1.CHINHANH@DB_LINK_PUN_CN1;
                            THONG_TIN_CN_VAL:=1;
                        END;
                    ELSE
                        BEGIN
                            SELECT COUNT(*) INTO NOT_EMPTY
                            FROM C##PUN_CN2.CHINHANH
                            WHERE MACN = V_MACN;
                            IF NOT_EMPTY = 1 THEN
                                BEGIN
                                    SELECT  MACN,  TENCN,  THANHPHO
BULK COLLECT INTO V_CHINHANH
                                    FROM C##PUN_CN2.CHINHANH;
                                    THONG_TIN_CN_VAL:=1;
                                END;
                            END IF;
                        END;
                    END IF;
                END;
            END IF;
```

```
        END;

    PROCEDURE HIEN_THI_PHONG_MOT_CN IS
        BEGIN
            SELECT COUNT(*) INTO NOT_EMPTY
            FROM    C##PUN_CN1.PHONG@DB_LINK_PUN_CN1    P    JOIN
C##PUN_CN1.LOAIPHONG@DB_LINK_PUN_CN1    LP    ON    P.MALOAIPH    =
LP.MALOAIPH
            WHERE TINHTRANG = 'Trống' AND MACN = V_MACN;
            IF NOT_EMPTY <> 0 THEN
                BEGIN
                    SELECT  DISTINCT  MAPH,  TENLOAIPH,  DONGIA,
SONGUOI, MACN BULK COLLECT INTO V_PHONG_CHINHANH
                    FROM  C##PUN_CN1.PHONG@DB_LINK_PUN_CN1  P  JOIN
C##PUN_CN1.LOAIPHONG@DB_LINK_PUN_CN1    LP    ON    P.MALOAIPH    =
LP.MALOAIPH
                    WHERE TINHTRANG ='Trống' AND MACN = 'CN0001'
                    ORDER BY MAPH ASC;
                END;
            ELSE
                BEGIN
                    SELECT COUNT(*) INTO NOT_EMPTY
                    FROM       C##PUN_CN2.PHONG       P       JOIN
C##PUN_CN2.LOAIPHONG LP ON P.MALOAIPH = LP.MALOAIPH
                    WHERE TINHTRANG = 'Trống' AND MACN = V_MACN;
                    IF NOT_EMPTY <> 0 THEN
                        BEGIN
                            SELECT   DISTINCT   MAPH,   TENLOAIPH,
DONGIA, SONGUOI, MACN BULK COLLECT INTO V_PHONG_CHINHANH
                            FROM    C##PUN_CN2.PHONG    P    JOIN
C##PUN_CN2.LOAIPHONG LP ON P.MALOAIPH = LP.MALOAIPH
                            WHERE TINHTRANG = 'Trống' AND MACN =
'CN0002'
                            ORDER BY MAPH ASC;
                        END;
                    END IF;
                END;
            END IF;
        END;

    PROCEDURE HIEN_THI_PHONG_BA_CN IS
        BEGIN
            SELECT MAPH, TENLOAIPH, DONGIA, SONGUOI, MACN BULK
COLLECT INTO V_PHONG_CHINHANH
            FROM (
```

```
                    SELECT DISTINCT MAPH, TENLOAIPH, DONGIA,
SONGUOI, MACN
                    FROM C##PUN_CN1.PHONG@DB_LINK_PUN_CN1 P JOIN
C##PUN_CN1.LOAIPHONG@DB_LINK_PUN_CN1 LP ON P.MALOAIPH =
LP.MALOAIPH
                    WHERE TINHTRANG = 'Trống' AND MACN = 'CN0001'
                    UNION
                    SELECT DISTINCT MAPH, TENLOAIPH, DONGIA,
SONGUOI, MACN
                    FROM C##PUN_CN2.PHONG P JOIN
C##PUN_CN2.LOAIPHONG LP ON P.MALOAIPH = LP.MALOAIPH
                    WHERE TINHTRANG = 'Trống' AND MACN = 'CN0002'
                )
                ORDER BY MACN ASC, MAPH ASC;
                SELECT COUNT(*) INTO NOT_EMPTY
                FROM V_PHONG_CHINHANH;
        END;

    PROCEDURE SHOW_INFORMATION_ROOM IS
        CHINHANH_INDEX PLS_INTEGER;
        PHONG_CHINHANH_INDEX PLS_INTEGER;
    BEGIN
        CHINHANH_INDEX := V_CHINHANH.FIRST;
        LOOP
            EXIT WHEN CHINHANH_INDEX IS NULL;
            DBMS_OUTPUT.PUT_LINE(V_CHINHANH(CHINHANH_INDEX).MACN
||'          '|| V_CHINHANH(CHINHANH_INDEX).TENCN||'          '||
V_CHINHANH(CHINHANH_INDEX).THANHPHO);

            IF V_MACN IS NULL THEN
                BEGIN
                    HIEN_THI_PHONG_BA_CN;
                END;
            ELSE
                BEGIN
                    HIEN_THI_PHONG_MOT_CN;
                END;
            END IF;

            FOR PHONG_CHINHANH_INDEX IN 1 ..
V_PHONG_CHINHANH.COUNT
            LOOP
                DBMS_OUTPUT.PUT_LINE('
'||V_PHONG_CHINHANH(PHONG_CHINHANH_INDEX).MAPH||'
'||V_PHONG_CHINHANH(PHONG_CHINHANH_INDEX).TENLOAIPH||'
```

```
'||V_PHONG_CHINHANH(PHONG_CHINHANH_INDEX).DONGIA||'
'||V_PHONG_CHINHANH(PHONG_CHINHANH_INDEX).SONGUOI);
            END LOOP;
            CHINHANH_INDEX := V_CHINHANH.NEXT(CHINHANH_INDEX);
        END LOOP;
    END;
BEGIN
    THONG_TIN_CN_VAL :=0;
    NOT_EMPTY := 0;
    THONG_TIN_CN;
    IF THONG_TIN_CN_VAL = 0 THEN
        BEGIN
            DBMS_OUTPUT.PUT_LINE('Khong  ton  tai  chi  nhanh  can
tim');
        END;
    ELSE
        BEGIN
            SHOW_INFORMATION_ROOM;
            IF NOT_EMPTY = 0 THEN
                BEGIN
                    DBMS_OUTPUT.PUT_LINE('Khong  ton  tai  du  lieu
phong con trong');
                END;
            END IF;
        END;
    END IF;
END;
```

2. Function :

a. *In ra tổng các hóa đơn của một khách hàng sử dụng tại tất cả chi nhánh với khách hàng có MAKH được truyền vào*

- Tên: **SUM_KHACHHANG**
  Trên bảng: HOADON, KHACHHANG
- Mã PL/SQL:

```
create or replace type t_record as object (
   V_MACN varchar2(20),
   V_MAKH varchar2(20),
    V_NGAYTT date,
   V_TONGTIEN number
);
create or replace type t_table as table of t_record;
```

```
create or replace function SUM_DOANHTHU ( V_MACN in VARCHAR2)
return t_table as v_ret   t_table;
begin
   select t_record(MACN, SUM(TONGTIEN) AS DOANHTHU) bulk collect into v_ret
   FROM (
IF(V_MACN='CN02') THEN
        SELECT MACN, SUM(TONGTIEN) AS DOANHTHU
        from C##CN02.HOADON@BTL1,C##CN02.CT_PHIEUTHUEPHONG@BTL1,
C##CN02.PHONG@BTL1
        WHERE HOADON.MAPTP = CT_PHIEUTHUEPHONG.MAPTP and
CT_PHIEUTHUEPHONG.MAPH
        ELSE

        SELECT MACN, MAKH, NGAYTT, TONGTIEN
        from C##CN03.HOADON,C##CN03.CT_PHIEUTHUEPHONG, C##CN03.PHONG
        WHERE HOADON.MAPTP = CT_PHIEUTHUEPHONG.MAPTP and
CT_PHIEUTHUEPHONG.MAPH


END IF;
     )
   where MAKH = V_MAKH
   return v_ret;
end;
```

Bước thực hiện:

[1]: Nhập vào mã khách hàng muốn xem tổng hóa đơn tại tất cả các chi nhánh.

[2]: Sau đó tìm kiếm và tính tổng các hóa đơn tại chi nhánh thứ nhất, thứ hai, thứ ba.

[3]: Tiến hành in ra tổng hóa đơn của khách hàng.

b. Tính doanh thu của chi nhánh được nhập vào thông qua mã chi nhánh

- Tên: **SUM_DOANHTHU**
  Trên bảng: HOADON
- Mã PL/SQL:

```
create or replace type t_record as object (
    V_MACN varchar2(20),
    V_MAKH varchar2(20),
     V_NGAYTT date,
    V_TONGTIEN number
);
create or replace type t_table as table of t_record;

create or replace function SUM_DOANHTHU ( V_MACN in VARCHAR2)
```

```
return t_table as v_ret   t_table;
begin
    select t_record(MACN, SUM(TONGTIEN) AS DOANHTHU) bulk
collect into v_ret
    FROM (
IF(V_MACN='CN02') THEN
            SELECT MACN, SUM(TONGTIEN) AS DOANHTHU
            from
C##CN02.HOADON@BTL1,C##CN02.CT_PHIEUTHUEPHONG@BTL1,
C##CN02.PHONG@BTL1
            WHERE HOADON.MAPTP = CT_PHIEUTHUEPHONG.MAPTP and
CT_PHIEUTHUEPHONG.MAPH
            ELSE
            SELECT MACN, MAKH, NGAYTT, TONGTIEN
            from C##CN03.HOADON,C##CN03.CT_PHIEUTHUEPHONG,
C##CN03.PHONG
            WHERE HOADON.MAPTP = CT_PHIEUTHUEPHONG.MAPTP and
CT_PHIEUTHUEPHONG.MAPH

END IF;
        )
    where MAKH = V_MAKH
    return v_ret;
end;
```

Bước thực hiện:

[1]: Nhập vào mã chi nhánh muốn tính doanh thu

[2]: Sau đó tìm kiếm đúng chi nhánh đã nhập để lấy các thông tin hóa đơn tại chi nhánh đó

[3]: Tiến hành in ra tổng doanh thu tại chi nhánh.

3. Ràng buộc toàn vẹn

❖ Tính tiền hóa đơn tại khách sạn khi sử dụng thuê phòng và dịch vụ:
Tiền hóa đơn = (Ngày trả phòng – Ngày nhận phòng + 1)*Giá phòng * (1+ Phụ thu) + Số lượng dịch vụ*Giá dịch vụ

• RBVT do thuộc tính tổng hợp

• Bối cảnh: HOADON, CT_PHIEUTHUEPHONG, PHONG, LOAIPHONG, DICHVU, CT_PHIEUDICHVU

• ∀ h ∈ HOADON, c ∈ CT_PHIEUTHUEPHONG, c1 ∈ CT_PHIEUDICHVU, ph ∈ PHONG, l ∈ LOAIPHONG, d ∈ DICHVU :

(h.MAPTP = c.MAPTP ∧ c1.MAPDV = h.MAPDV) ∧ (c.MAPH = ph.MAPH) ∧ (ph.MALOAIPH = l.MALOAIPH) ∧ (c1.MADV = d.MADV) → h.TongTien = (c.NgayTPTT – c.NgayNP + 1) *** l.DonGia * (1 + c.PhuThu) + c1.SLDV * d.GiaDV**

• Bảng tầm ảnh hưởng:

R21	Thêm	Xóa	Sửa
CT_PHIEUTHUEPHONG	+	+	+(MAPTP, MAPH)
HOADON	+	-	+(TongTien)
PHONG	-	+	+(MALOAIPH)
LOAIPHONG	-	+	+(DonGia)
DICHVU	-	+	+(TienDV)
CT_PHIEUDICHVU	+	+	+(MAPDV, MADV)

❖ Phòng thuộc một loại phòng nhất định

• RBTV tham chiếu

• Bối cảnh: PHONG, LOAIPHONG

• ∀ p ∈ PHONG, ∃ lp∈ LOAIPHONG: p. MALOAIPH = lp. MALOAIPH

• Bảng tầm ảnh hưởng:

R14	Thêm	Xóa	Sửa
PHONG	+	-	+(MALOAIPH)
LOAIPHONG	-	+	-(*)

III. Demo các mức cô lập (ISOLATION LEVEL) trong môi trường phân tán

- ❖ Demo các mức cô lập trên hệ quản trị Oracle trong môi trường phân tán
- ❖ Đưa ra trường hợp xảy ra và cách giải quyết với từng trường hợp gây mất tính nhất quán

1. Non – repeatable Read:

- 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.

a. Tình huống: Khách hàng Phạm Thị Ngọc Nam (makh = 'KH0011') của muốn đặt vé DamSen's Park tại Hotel vì giá vé rẻ hơn bên ngoài 10%. Khách hàng gọi điện cho nhân viên tại chi nhánh Hồ Chí Minh và kiểm tra vé thì thấy giá vé lúc này là 320.000, Nam quyết định đặt vé tại đây. Thì ngay lúc này, khách sạn nhận được cuộc gọi bởi đơn vị DamSen'sPark và tăng mức giá của vé thêm 5%, nhân viên bên chi nhánh 'Huế' nhận được thông báo sớm nhất và tiến hành chỉnh sửa thông tin. Khi xuất hóa đơn cho khách hàng, thì nhân viên và Nam đã nhận thấy sự thay đổi trong giá vé.

b. Mô tả:

Session 1	Session 2	Explanation
SELECT * FROM DICHVU; MADV TENDV GIADV ---- -------- ----- …… …… …… DV0023 Đặt vé khu vui chơi 320000 …… …… ……	No action.	Session 1 thực hiện câu lệnh truy vấn tìm kiếm giá dịch vụ khu vui chơi của DamSen'Park. Thì thấy giá chương trình "Đặt vé khu vui chơi" có giá là 320,000.

No action.	BEGIN
	UPDATE_GIADV_THEOPHANTRAM('DV0023', 0.05);
	END;
	
	PL/SQL procedure successfully completed.
		Session 2 sử dụng mức cô lập mặc định trong Oracle là 'SET TRANSACTION ISOLATION LEVEL READ COMMITTED' và thực hiện lệnh cập nhật theo yêu cầu của đơn vị là tăng giá thêm 5%.

Session 1	Session 2	Explanation
No action.	COMMIT; `Commit completed.`	Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.
INSERT INTO PHIEUDICHVU(TIENDV) VALUES(0); `1 row insert.`	No action.	Session 1 sau khi tra cứu giá vé đã tiến hành lập phiếu dịch vụ cho khách hàng có mã khách hàng là 'KH0011' bởi nhân viên có mã là 'NV0005'.

SELECT *
FROM PHIEUDICHVU
WHERE TIENDV = 0;
 	
`MAPDV TIENDV`
`---- ------`
`PDV0033 0`
	No action.	Session 1 kiểm tra lập phiếu thành công.

INSERT INTO CT_PHIEUDICHVU(MAPDV, MADV, SLDV, NGAYSD) VALUES('MPDV0033', 'DV0023' , 2, SYSDATE); `1 row insert.`	No action.	Session 1 tiếp tục thực hiện thao tác thêm vào chi tiết phiếu dịch vụ có mã là mã phiếu của khách hàng vừa đăng ký, mã dịch vụ là 'DV0023' và số lượng dịch vụ là 2.

SELECT *
FROM CT_PHIEUDICHVU
WHERE MAPDV = 'MPDV0033';
 	
`MAPDV MADV SLDV NGAYSD`
`---- ---- ----- ------`
`PDV0033 DV0023 2 07/12/2021`
	No action.	Session 1 kiểm tra thêm chi tiết phiếu dịch vụ thành công.

BEGIN
THEM_HOADON_PDV('KH0011', 'NV0005', 'MPDV0033');
END;
 	
`PL/SQL procedure successfully completed.`
	No action.	Session 1 tiến hành lập hóa đơn cho khách hàng với mã khách hàng là 'KH0011' và mã nhân viên là 'NV0005' với mã phiếu dịch vụ là 'MPDV0033'.

Session 1	Session 2	Explanation
SELECT * FROM HOADON WHERE MAPDV = 'MPDV0033' MAHD TONGTIEN NGAYTT MANV ---- -------- ------ ---- HD0033 672000 07/12/2021 NV0005 MAKH MAPTP MAPDV ---- ---- --- KH0018 NULL MPDV0033	No action.	Session 1 thực hiện câu lệnh truy vấn thông tin hóa đơn của mã phiếu dịch vụ khách hàng Nam đang sử dụng. Session 1 tiến hành xuất hóa đơn với mã hóa đơn là 'HD0033'. Và nhận thấy tiền dịch vụ của chương trình "Đặt vé khu vui chơi" đã tăng từ 320000 lên 336000 so với ban đầu dẫn đến chi phí của khách hàng Nam phải trả cao hơn so với dự tính.

→ Unrepeatable Read đã xảy ra

c. Nguyên nhân và giải pháp:

- Vấn đề: Session 1 thực hiện truy vấn thông tin dịch vụ, ngay lúc này session 2 cập nhật giá dịch vụ tương ứng, sau đó session 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.
- Nguyên nhân: vì mức cô lập của Session 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 session khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.
- Giải pháp: đổi mức cô lập ở Session 1 thành **SERIALIZABLE** thay vì **READ COMMITTED** trước khi thực thi.
- Xử lý Non – repeatable Read bằng Serializable:

Session 1	Session 2	Explanation
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; Transaction ISOLATION succeeded	No action.	Session 1 thiết lập lức cô lập Serializable.
SELECT * FROM DICHVU; MADV TENDV GIADV ---- -------- ----- …… …… …… DV0023 Đặt vé khu vui chơi 320000 …… …… ……	No action.	Session 1 thực hiện câu lệnh truy vấn tìm kiếm giá dịch vụ khu vui chơi của DamSen'Park thì thấy giá chương trình "Đặt vé khu vui chơi" có giá là 320,000.

Session 1	Session 2	Explanation

No action.	BEGIN
	UPDATE_GIADV_THEO PHANTRAM('DV0023', 0.05);
	END;
	
	PL/SQL procedure successfully completed.
		Session 2 sử dụng mức cô lập mặc định trong Oracle là 'SET TRANSACTION ISOLATION LEVEL READ COMMITTED' và thực hiện lệnh cập nhật theo yêu cầu của đơn vị là tăng giá thêm 5%.

No action.	COMMIT; Commit completed.	Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.
INSERT INTO PHIEUDICHVU(TIENDV) VALUES(0); 1 row insert.	No action.	Session 1 sau khi tra cứu giá vé đã tiến hành lập phiếu dịch vụ cho khách hàng có mã khách hàng là 'KH0011' bởi nhân viên có mã là 'NV0005'.
SELECT * FROM PHIEUDICHVU WHERE TIENDV = 0; MAPDV TIENDV ---- ------ PDV0033 0	No action.	Session 1 kiểm tra lập phiếu thành công.
INSERT INTO CT_PHIEUDICHVU(MAPDV, MADV, SLDV, NGAYSD) VALUES('MPDV0033', 'DV0023' , 2, SYSDATE); 1 row insert.	No action.	Session 1 tiếp tục thực hiện thao tác thêm vào chi tiết phiếu dịch vụ có mã là mã phiếu của khách hàng vừa đăng ký, mã dịch vụ là 'DV0023' và số lượng dịch vụ là 2.
SELECT * FROM CT_PHIEUDICHVU WHERE MAPDV = 'MPDV0033'; MAPDV MADV SLDV NGAYSD ---- ---- ----- ------ PDV0033 DV0023 2 07/12/2021	No action.	Session 1 kiểm tra thêm chi tiết phiếu dịch vụ thành công.

Session 1	Session 2	Explanation

BEGIN
THEM_HOADON_PDV('KH0011', 'NV0005', 'MPDV0033');
END;
 	
PL/SQL procedure successfully completed.
	No action.	Session 1 tiến hành lập hóa đơn cho khách hàng với mã khách hàng là 'KH0011' và mã nhân viên là 'NV0005' với mã phiếu dịch vụ là 'MPDV0033'.

SELECT * FROM HOADON WHERE MAPDV = 'MPDV0033' MAHD TONGTIEN NGAYTT MANV ---- -------- ------ ---- HD0033 640000 07/12/2021 NV0005 MAKH MAPTP MAPDV ---- ---- --- KH0018 NULL MPDV0033		Session 1 thực hiện câu lệnh truy vấn thông tin hóa đơn của mã phiếu dịch vụ khách hàng Nam đang sử dụng. Session 1 tiến hành xuất hóa đơn với mã hóa đơn là 'HD0033'. Và nhận thấy tiền dịch vụ của chương trình "Đặt vé khu vui chơi" không đổi, khách hàng Thiện đăng ký dịch vụ và thanh toán thành công.
COMMIT; Commit completed.		Session 1 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.

Hàm hỗ trợ:

1. Hàm UPDATE_GIADV_THEOPHANTRAM

```
CREATE OR REPLACE PROCEDURE UPDATE_GIADV_THEOPHANTRAM(V_MADV
DICHVU.MADV%TYPE,
                                                      V_PHANTRAM
NUMBER
                                                      )
IS
BEGIN
    UPDATE C##CN02.DICHVU@BTL1
    SET GIADV = GIADV *(1 + V_PHANTRAM)
    WHERE MADV = V_MADV;
    UPDATE C##CN03.DICHVU
    SET GIADV = GIADV *(1 + V_PHANTRAM)
    WHERE MADV = V_MADV;
END;
```

2. Hàm THEM_HOADON

```
CREATE OR REPLACE PROCEDURE THEM_HOADON_PDV(V_MAKH
KHACHHANG_HIENTHI.MAKH%TYPE,
                                            V_MANV NHANVIEN.MANV%TYPE,
                                            V_PDV
PHIEUDICHVU.MAPDV%TYPE
                                            )
IS
    V_TIENDV PHIEUDICHVU.TIENDV%TYPE;
BEGIN
    SELECT SUM(GIADV * SLDV) INTO V_TIENDV
    FROM DICHVU DV JOIN CT_PHIEUDICHVU CTPDV ON DV.MADV = CTPDV.MADV
    WHERE MAPDV = V_PDV;

    UPDATE PHIEUDICHVU
    SET TIENDV = V_TIENDV
    WHERE MAPDV = V_PDV;

    INSERT INTO HOADON(TONGTIEN,NGAYTT,MANV,MAKH,MAPTP,MAPDV)
VALUES(V_TIENDV, SYSDATE, V_MANV, V_MAKH, NULL, V_PDV);
END;
```

2. **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.

a. Tình huống: Quản lý ở chi nhánh Huế tiến hành tra cứu thông tin chi tiết tình trạng phòng ở tất cả các chi nhánh để thống kê số lượng phòng trống để tìm hiểu và đưa ra chiến lược về tình hình doanh thu sắp tới. Cùng lúc đó, khách hàng Hoàng Minh Hùng có mã khách hàng là KH0019 đã tiến hành trả phòng đã đặt trước khi tới khách sạn do tình hình dịch bệnh ngày càng căng thẳng. Lúc này, quản lý tra cứu lại thông tin và nhận thấy dữ liệu bên trong đã thay đổi.

b. Mô tả:

Session 1	Session 2	Explanation

BEGIN
SHOW_INFORMATION_VACANCIES (NULL);
END;
MACN TENCN THANHPHO
------- -------- --------
CN0001 CN Hà Nội Hà Nội
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0006 Standard Room 800000 3
MP0007 Superior Room 1500000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0002 CN Huế Huế
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0003 Superior Room 1500000 3
MP0004 Deluxe Room 2000000 3
MP0005 Deluxe Room 2000000 3
MP0006 Deluxe Room 2000000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0003 CN Hồ Chí Minh HCM
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0005 Connecting Room 2000000 3
MP0006 Deluxe Room 2000000 3
MP0007 Connecting Room 2000000 3
MP0008 Standard Room 2000000 3
	No action.	Session 1 thực hiện thủ tục SHOW_INFORMATION_VACANCIES và tìm kiếm các phòng có tình trạng là trống.

No action.	BEGIN
	XOA_PTP (‘MPTP0009’);
	END;
	
	PL/SQL procedure successfully completed.
		Session 2 thực hiện giao tác xóa phiếu thuê phòng của khách hàng có makh = ‘KH0019’.
		Hệ thống tự động cập nhật lại tình trạng phòng khi xóa các dữ liệu trong CT_PHIEUTHUEPHONG.
		Trong thủ tục XOA_PTP, khi tiến hành xóa phiếu đặt phòng, thì đồng thời cũng sẽ hóa đơn tạm thời. Lúc này, Session 2 thực hiện thành công và kết thúc.

Session 1	Session 2	Explanation
	COMMIT; `Commit completed.`	Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh 'COMMIT'.

BEGIN
SHOW_INFORMATION_VACANCIES (NULL);
END;
`MACN TENCN THANHPHO`
`------- -------- --------`
`CN0001 CN Hà Nội Hà Nội`
`MAPH TENLOAIPH DONGIA SONGUOI`
`------- -------- ------ ------`
`MP0006 Standard Room 800000 3`
`MP0007 Superior Room 1500000 3`
 	
`MACN TENCN THANHPHO`
`------- -------- --------`
`CN0002 CN Huế Huế`
`MAPH TENLOAIPH DONGIA SONGUOI`
`------- -------- ------ ------`
`MP0003 Superior Room 1500000 3`
`MP0004 Deluxe Room 2000000 3`
`MP0005 Deluxe Room 2000000 3`
`MP0006 Deluxe Room 2000000 3`
 	
`MACN TENCN THANHPHO`
`------- -------- --------`
`CN0003 CN Hồ Chí Minh HCM`
`MAPH TENLOAIPH DONGIA SONGUOI`
`------- -------- ------ ------`
`MP0004 Superior Room 1500000 3`
`MP0005 Connecting Room 2000000 3`
`MP0006 Deluxe Room 2000000 3`
`MP0007 Connecting Room 2000000 3`
`MP0008 Standard Room 2000000 3`
	No action.	Session 1 thực hiện lại giao tác tra cứu thì thông tin phòng bên chi nhánh 3 đã thay đổi.
		Mã phòng 'MP0004' đã xuất hiện trong dữ liệu bảng còn trống.

→ Phantom Read đã xảy ra.

c. Nguyên nhân và giải pháp:

- Vấn đề: Session 1 thực hiện truy vấn thông tin tình trạng phòng trống, ngay lúc này session 2 xóa phiếu thuê phòng trước và thay đổi tình trạng phòng tương ứng, sau đó session 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.
- Nguyên nhân: vì mức cô lập của session 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 session khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.
  Giải pháp: đổi mức cô lập ở session1 thành **SERIALIZABLE** thay vì **READ COMMITTED**.
- Xử lý Phantom Read bằng Serializable:

Session 1	Session 2	Explanation
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; Transaction ISOLATION succeeded	No action.	Session 1 thiết lập lức cô lập Serializable.

BEGIN
SHOW_INFORMATION_VACANCIES (NULL);
END;
MACN TENCN THANHPHO
------- -------- --------
CN0001 CN Hà Nội Hà Nội
MẠPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0006 Standard Room 800000 3
MP0007 Superior Room 1500000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0002 CN Huế Huế
MẠPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0003 Superior Room 1500000 3
MP0004 Deluxe Room 2000000 3
MP0005 Deluxe Room 2000000 3
MP0006 Deluxe Room 2000000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0003 CN Hồ Chí Minh HCM
MẠPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0005 Connecting Room 2000000 3
MP0006 Deluxe Room 2000000 3
MP0007 Connecting Room 2000000 3
MP0008 Standard Room 2000000 3
	No action.	Session 1 thực hiện thủ tục SHOW_INFORMATION _VACANCIES và tìm kiếm các phòng có tình trạng là trống.

Session 1	Session 2	Explanation

No action.	BEGIN
	XOA_PTP (‘MPTP0009’);
	END;
	
	PL/SQL procedure successfully completed.
		Session 2 thực hiện giao tác xóa phiếu thuê phòng của khách hàng có makh = ‘KH0019’.
		Hệ thống tự động cập nhật lại tình trạng phòng khi xóa các dữ liệu trong CT_PHIEUTHUEPHONG
		Trong thủ tục XOA_PTP, khi tiến hành xóa phiếu đặt phòng, thì đồng thời cũng sẽ hóa đơn tạm thời.
		Lúc này, Session 2 thực hiện thành công và kết thúc.

BEGIN
SHOW_INFORMATION_VACANCIES (NULL);
END;
MACN TENCN THANHPHO
------- -------- --------
CN0001 CN Hà Nội Hà Nội
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0006 Standard Room 800000 3
MP0007 Superior Room 1500000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0002 CN Huế Huế
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0003 Superior Room 1500000 3
MP0004 Deluxe Room 2000000 3
MP0005 Deluxe Room 2000000 3
MP0006 Deluxe Room 2000000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0003 CN Hồ Chí Minh HCM
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0005 Connecting Room 2000000 3
MP0006 Deluxe Room 2000000 3
MP0007 Connecting Room 2000000 3
MP0008 Standard Room 2000000 3
	No action.	Session 1 thực hiện thủ tục SHOW_INFORMATION_VACANCIES và tìm kiếm các phòng có tình trạng phòng trống một lần nữa. Lúc này, thông tin không bị thay đổi. Quản lý xác nhận thông tin chính xác và tiến hành lập báo cáo, xây dựng chiến lược cho khách sạn.

COMMIT; Commit completed.		Session 1 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’* quá trình truy xuất thông tin.

Session 1	Session 2	Explanation

BEGIN
SHOW_INFORMATION_VACANCIES (NULL);
END;
MACN TENCN THANHPHO
------- -------- --------
CN0001 CN Hà Nội Hà Nội
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0006 Standard Room 800000 3
MP0007 Superior Room 1500000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0002 CN Huế Huế
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0003 Superior Room 1500000 3
MP0004 Deluxe Room 2000000 3
MP0005 Deluxe Room 2000000 3
MP0006 Deluxe Room 2000000 3
 	
MACN TENCN THANHPHO
------- -------- --------
CN0003 CN Hồ Chí Minh HCM
MAPH TENLOAIPH DONGIA SONGUOI
------- -------- ------ ------
MP0004 Superior Room 1500000 3
MP0005 Connecting Room 2000000 3
MP0006 Deluxe Room 2000000 3
MP0007 Connecting Room 2000000 3
MP0008 Standard Room 2000000 3
	No action.	Session 1 thực hiện lại giao tác tra cứu thì thông tin phòng bên chi nhánh 3 đã thay đổi.
		Mã phòng 'MP0004' đã xuất hiện trong dữ liệu bảng còn trống.

Hàm hỗ trợ:

3. Dữ liệu hỗ trợ:

```
INSERT INTO PHIEUTHUEPHONG(MAPTP,TIENTP) VALUES('MPTP0009',0);
INSERT INTO CT_PHIEUTHUEPHONG
(MAPTP,MAPH,NGAYNP,NGAYTPTT,PHUTHU,SLNGUOI) values
('MPTP0009','MP0004',SYSDATE,SYSDATE + 2, 0, 2);
INSERT INTO HOADON (MAHD, TONGTIEN,NGAYTT,MANV,MAKH,MAPTP,MAPDV)
values ('HD0013', 0, NULL,'NV0008','KH0016','MPTP0009',NULL);
UPDATE PHONG SET TINHTRANG = 'Sử dụng' WHERE MAPH = 'MP0004';
```

4. Hàm XOA_PTP

```
CREATE OR REPLACE PROCEDURE XOA_PTP (V_MAPTP
PHIEUTHUEPHONG.MAPTP%TYPE)
IS
BEGIN
    DELETE FROM PHIEUTHUEPHONG WHERE MAPTP = V_MAPTP;
```

```
    DELETE FROM HOADON WHERE MAPTP =  V_MAPTP;
END;
```

5. Trigger Xoa_PTP

```
CREATE OR REPLACE TRIGGER XOA_PTP
BEFORE DELETE ON PHIEUTHUEPHONG
FOR EACH ROW
BEGIN
    UPDATE PHONG
    SET TINHTRANG = 'Trống'
    WHERE MAPH IN (SELECT MAPH
                   FROM CT_PHIEUTHUEPHONG
                   WHERE MAPTP =  :OLD.MAPTP
                );
    DELETE FROM CT_PHIEUTHUEPHONG WHERE MAPTP = :OLD.MAPTP;

END;
```

3. Lost update

a. **Mô tả tình huống**: Vào ngày 5 tháng 6 hằng năm, hệ thống khách sạn thường thực hiện việc tăng lương cho nhân viên nhằm khuyến khích nhân viên làm việc, gắn bó lâu dài với công ty. Tình huống đặt ra tại đây là, khi quản lí A thuộc chi nhánh 1 là chi nhánh tổng đang cập nhật tăng mức lương cho Nhân viên A tại chi nhánh 2. Trong cùng lúc đó, tại chi nhánh 2, quản lí B cũng đồng thời thực hiện việc cập nhật giảm cho mức lương của Nhân viên A vì Nhân viên A vi phạm quy định của khách sạn. Chính vì thế. Khi vừa cập nhật xong kết quả tăng lương nhưng quản lí A lại thấy kết quả hiển thị không chính xác.

b. **Bảng mô tả Lost_update**

Chi nhánh 2	Chi nhánh 3	Giải thích
SELECT * FROM C##CN03.NHANVIEN@BTL1 where MANV="NV0001"; MaNV HoTen Luong ---------- NV0001 A 4000000	Không thực hiện.	Chi nhánh 2 thực hiện truy vấn dữ liệu của nhân viên ‘NV0001’ .
UPDATE C##CN03.NHANVIEN@BTL1 SET LUONG=5000000 WHERE MANV="NV0001";	Không thực hiện.	Chi nhánh 2 thực hiện cập nhật cập nhật lương của ‘NV0001’ thành 5000000

Chi nhánh 2	Chi nhánh 3	Giải thích
Không thực hiện.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED;	Chi nhánh 3 bắt đầu thiết lập mức cô lập READ COMMITTED.
Không thực hiện.	SELECT * FROM C##CN03.NHANVIEN where MANV="NV0001"; MaNV HoTen Luong ------------------ NV0001 A 4000000	CHI NHÁNH 3 truy vấn đến LƯƠNG của nhân viên khi CHI NHÁNH 2 cập nhật và chưa commit dữ liệu.
Không thực hiện.	UPDATE C##CN03.NHANVIEN SET LUONG=3000000 WHERE MANV="NV0001";	CHI NHÁNH 3 cập nhật lương cho 'NV0001' nhưng không thành công do giao tác 1 đang giữ khóa.
Commit.	Không thực hiện	CHI NHÁNH 2 thực hiện thành công và kết thúc.
Không thực hiện	1 row updated	CHI NHÁNH 3 cập nhật thành công
Không thực hiện	Commit.	CHI NHÁNH 3 cũng thực hiện thành công và kết thúc.
SELECT * FROM C##CN03.NHANVIEN@BTL1 where MANV="NV0001"; MaNV HoTen Luong ------------------ NV0001 A 3000000		CHI NHÁNH 2 thực hiện truy vấn thông tin của NHANVIEN 'NV0001' nhưng giá trị không đúng với giá trị đã cập nhật.

❖ Nguyên nhân và giải pháp

- Vấn đề xảy ra: khi Chi nhánh 2 cập nhật LƯƠNG cho nhân viên NV0001 nhưng sau khi xem lại thì không đúng dữ liệu.
- Nguyên nhân: Chi nhánh 3 cập nhật dữ liệu trước khi chi nhánh 2 tiến hành commit hoặc rollback vì vậy dữ liệu hiển thị tại Chi nhánh 3 commit sau khi Chi nhánh 3 update nó sẽ chờ khóa của Chi nhánh 2 và không thể thực hiện hành động khác nên không thể commit, trước khi chi nhánh 2 commit.

- Giải pháp: sử dụng câu lệnh “set transaction isolation level serializable” thay cho câu lệnh “set transaction isolation level read committed”

Chi nhánh 2	Chi nhánh 3	Giải thích
SELECT * FROM C##CN03.NHANVIEN@BTL1 where MANV="NV0001"; MaNV HoTen Luong ------------------ NV0001 A 4000000	Không thực hiện.	Chi nhánh 2 thực hiện truy vấn dữ liệu của nhân viên ‘NV0001’.
UPDATE C##CN03.NHANVIEN@BTL1 SET LUONG=5000000 WHERE MANV="NV0001";	Không thực hiện.	Chi nhánh 2 thực hiện cập nhật cập nhật lương của ‘NV0001’ thành 5000000.
Không thực hiện.	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	Chi nhánh 3 bắt đầu giao tác 3 và thiết lập mức cô lập SERIALIZABLE.
Không thực hiện.	SELECT * FROM C##CN03.NHANVIEN where MANV="NV0001"; MaNV HoTen Luong ------------------ NV0001 A 4000000	CHI NHÁNH 3 truy vấn đến lương khi CHI NHÁNH 2 chưa cập nhật và commit dữ liệu.
Không thực hiện.	UPDATE C##CN03.NHANVIEN SET LUONG=3000000 WHERE MANV="NV0001";	CHI NHÁNH 3 cập nhật lương cho ‘NV0001’ nhưng không thành công do CHI NHÁNH 2 đang giữ khóa.
Commit.	Trạng thái chờ	CHI NHÁNH 2 thực hiện commit thành công và kết thúc.

Chi nhánh 2	Chi nhánh 3	Giải thích
Không thực hiện	Báo lỗi: "ORA-08177: can't serialize access for this transaction."	Câu lệnh 'COMMIT' đã kết thúc phiên làm việc của Chi nhánh 2 và thay đổi dữ liệu trong NHÂN VIÊN nên CHI NHÁNH 3 không thực hiện thành công với lỗi 'ORA-08177'. Nguyên nhân vì CHI NHÁNH 3 đã cập nhật lương sau khi CHI NHÁNH 2 bắt đầu.
SELECT * FROM C##CN03.NHANVIEN@BTL1 where MANV="NV0001"; MaNV HoTen Luong ---------- NV0001 A 5000000	Không thực hiện	CHI NHÁNH 2 thực hiện truy vấn thông tin của NHÂN VIÊN 'NV0001' và thấy lương mới đã được cập nhật.

4. Deadlock

a. **Mô tả tình huống**: 2 nhân viên quản lí tiến hành cập nhật giá đơn giá của phòng đôi cùng 1 lúc. Khi đó xảy ra deadlock.

b. Bảng mô tả Deadlock

Chi nhánh 2	Chi nhánh 3	Explanation
UPDATE C##CN03.LOAIPHONG@BTL1 SET DONGIA = 170000 WHERE TENLOAIPH = 'Standard Room'	Không thực hiện	Cập nhật giá cho tên loại phòng là phòng Standard, Chi nhánh 2 phải xin khóa ghi rồi tiến hành cập nhật.
1 row updated	UPDATE C##CN03.LOAIPHONG SET DONGIA = 190000 WHERE TENLOAIPH =	Cập nhật giá cho loại tên loại phòng là phòng Superior, Chi nhánh 3 phải xin khóa ghi rồi tiến hành cập nhật.

Chi nhánh 2	Chi nhánh 3	Explanation
	'Superior Room'	
UPDATE C##CN03.LOAIPHONG@BTL1 SET DONGIA = 180000 WHERE TENLOAIPH = 'Superior Room'	1 row updated	Cập nhật đơn giá cho loại phòng Superior, Chi nhánh 2 phải xin khóa ghi rồi tiến hành cập nhật. Nhưng Chi nhánh 3 đang giữ khóa phải chờ Chi nhánh 3 nhả khóa(commit transaction)
T1 bị treo	UPDATE C##CN03.LOAIPHONG SET DONGIA = 180000 WHERE TENLOAIPH = 'Standard Room'	Cập nhật đơn giá cho loại phòng Standard, Chi nhánh 3 phải xin khóa ghi rồi tiến hành cập nhật. Nhưng Chi nhánh2 đang giữ khóa phải chờ Chi nhánh 3 nhả khóa(commit transaction)
SQL Error: ORA-00060: deadlock detected while waiting for resource 00060. 00000 - "deadlock detected while waiting for resource"	T2 bị treo	Chi nhánh 2, 3 chờ khóa của nhau nên DBMS bị treo. Sau 1 khoảng thời gian(Time out) DBMS tạo ra exception
Rollback completed	Không thực hiện	Giải phóng Chi nhánh2
Không thực hiện	1 row updated	Chi nhánh 3 xin được khóa và hoàn thành giao tác

c. Nguyên nhân và giải pháp

- **Vấn đề xảy ra** : Chi nhánh 3 chờ Chi nhánh 2 giải phóng khóa ghi, Chi nhánh 2 lại chờ Chi nhánh 3 giải phóng khóa ghi. Kết quả dẫn đến hệ quản trị báo lỗi và hủy Chi nhánh 2.
- **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 Chi nhánh 2 rollback thì tại Chi nhasnh2, 3 được updated thành công. Hoặc để giải quyết deadlock, tạo 1 function lock_row để khóa hàm NOWAIT và trả về giá trị boolean để cho biết thành công hay thất bại. Mỗi lần UPDATE ta sử dụng hàm để kiểm tra Transaction có bị khóa bởi một giao tác nào khác không, nếu không có thì sẽ tiến

hành giữ khóa và tiến hành UPDATE cho đến khi mở khóa (commit hoặc rollback). Nếu như truyền sai tên tham số thì sẽ không giữ khóa.

Hàm hỗ trợ:

1. lock_row: để khóa hàm NOWAIT

```
set serveroutput on;
create or replace function lock_row(tenlp VARCHAR2)
return boolean
is
resource_busy exception;
pragma        exception_init(resource_busy, -54);
v_maloaiph       DANHSACHPHONG.TENLOAIPH%type;
begin
select TENLOAIPH into v_maloaiph
from DANHSACHPHONG
where TENLOAIPH = tenlp
for update nowait;
return true;
exception
when resource_busy then
return false;
end;
```

IV. Thực hiện tối ưu hóa truy vấn trên môi trường phân tán 1 câu truy vấn đơn giản

- ❖ Tự đề xuất một câu truy vấn đơn giản chưa được tối ưu
- ❖ EXPLAIN Query câu truy vấn đơn giản
- ❖ Tối ưu hóa câu truy vấn cục bộ, phân tán
- ❖ Viết lại câu Query trên môi trường phân tán, nhận xét

- Ta có lược đồ phân mảnh như sau:

❖ Quan hệ CHINHANH phân mảnh ngang chính theo thành phố:

CHINHANH1 = $\sigma_{ThanhPho = \text{'Hà Nội'}}$ CHINHANH

CHINHANH2 = $\sigma_{ThanhPho = \text{'Huế'}}$ CHINHANH

CHINHANH3 = $\sigma_{ThanhPho = \text{'HCM'}}$ CHINHANH

❖ Quan hệ PHONG, NHANVIEN, HOADON, PHIEUTHUEPHONG, PHIEUDICHVU phân mảnh ngang dẫn xuất như sau:

PHONG1 = PHONG $\ltimes_{MACN}$ CHINHANH1

PHONG2 = PHONG $\ltimes_{MACN}$ CHINHANH2

PHONG3 = PHONG $\ltimes_{MACN}$ CHINHANH3

NHANVIEN1 = NHANVIEN $\ltimes_{MACN}$ CHINHANH1

NHANVIEN2 = NHANVIEN $\ltimes_{MACN}$ CHINHANH2

NHANVIEN3 = NHANVIEN $\ltimes_{MACN}$ CHINHANH3

HOADON1 = HOADON $\ltimes_{MANV}$ NHANVIEN1

HOADON2 = HOADON $\ltimes_{MANV}$ NHANVIEN2

HOADON3 = HOADON $\ltimes_{MANV}$ NHANVIEN3

PHIEUTHUEPHONG1 = PHIEUTHUEPHONG $\ltimes_{MAPTP}$ HOADON1

PHIEUTHUEPHONG2 = PHIEUTHUEPHONG $\ltimes_{MAPTP}$ HOADON2

PHIEUTHUEPHONG3 = PHIEUTHUEPHONG $\ltimes_{MAPTP}$ HOADON 3

CT_PHIEUTHUEPHONG1 = CT_PHIEUTHUEPHONG $\ltimes_{MAPTP}$ PHIEUTHUEPHONG1

CT_PHIEUTHUEPHONG2 = CT_PHIEUTHUEPHONG $\ltimes_{MAPTP}$ PHIEUTHUEPHONG2

CT_PHIEUTHUEPHONG3 = CT_PHIEUTHUEPHONG $\ltimes_{MAPTP}$ PHIEUTHUEPHONG3

PHIEUDICHVU1 = PHIEUDICHVU $\ltimes_{MAPDV}$ HOADON1

PHIEUDICHVU2 = PHIEUDICHVU $\ltimes_{MAPDV}$ HOADON2

PHIEUDICHVU3 = PHIEUDICHVU $\ltimes_{MAPDV}$ HOADON 3

CT_PHIEUDICHVU1 = CT_PHIEUDICHVU $\ltimes_{MAPDV}$ PHIEUDICHVU1

CT_PHIEUDICHVU2 = CT_PHIEUDICHVU $\ltimes_{MAPDV}$ PHIEUDICHVU2

CT_PHIEUDICHVU3 = CT_PHIEUDICHVU $\ltimes_{MAPDV}$ PHIEUDICHVU3

❖ Quan hệ KHACHHANG được phân mảnh hỗn hợp như sau:

KHACHHANG1A = $\prod_{MAKH,TENKH,CMND,QUOCTICH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH1)

KHACHHANG1B = $\prod_{MAKH, TENKH, NGSINH, DIACHI, SDT, LOAIKH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH1)

KHACHHANG2A = $\prod_{MAKH,TENKH,CMND,QUOCTICH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH2)

KHACHHANG2B = $\prod_{MAKH, TENKH, NGSINH, DIACHI, SDT, LOAIKH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH2)

KHACHHANG3A = $\prod_{MAKH,TENKH,CMND,QUOCTICH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH3)

KHACHHANG3B = $\prod_{MAKH, TENKH, NGSINH, DIACHI, SDT, LOAIKH}$ (KHACHHANG $\ltimes_{MACN}$ CHINHANH3)

❖ Quan hệ LOAIPHONG VÀ DICHVU được nhân bản ở tất cả các chi nhánh

Truy vấn đơn giản Q : **Tìm thông tin khách hàng ở chi nhánh Huế có hóa đơn thanh toán vào tháng 7 trong đó tiền dịch vụ > 10000000 và có sử dụng dịch vụ Casino**

```
SELECT KH.MAKH, KH.TENKH
FROM HOADON HD , PHIEUDICHVU PDV, CT_PHIEUDICHVU CTPDV, DICHVU
DV, KHACHHANG KH, CHINHANH CN
WHERE HD.MAPDV = PDV.MAPDV AND EXTRACT(MONTH FROM HD.NGAYTT) =
7 AND PDV.TIENDV > 10000000
AND CTPDV.MAPDV = PDV.MAPDV AND DV.MADV = CTPDV.MADV AND TENDV
= 'Casino' AND KH.MAKH = HD.MAKH
 AND TENCN = 'CN Huế' AND CN.MACN = KH.MACN AND QUOCTICH =
'Việt Nam';
```

Các từ viết tắt được sử đụng trong bài :

Từ viết tắt	Từ đầy đủ
KH	KHACHHANG
CN	CHINHANH
KH_LT	KHACHHANG_LUUTRU
HD	HOADON
PDV	PHIEUDICHVU
CTPDV	CT_PHIEUDICHVU
DV	DICHVU

1. Chạy Explain query Q

- Kết quả của query Q

```
1  SELECT /*+ GATHER_PLAN_STATISTICS */ KH.MAKH, KH.TENKH
2  FROM HOADON HD , PHIEUDICHVU PDV, CT_PHIEUDICHVU CTPDV, DICHVU DV, KHACHHANG KH, CHINHANH CN
3  WHERE HD.MAPDV = PDV.MAPDV AND EXTRACT(MONTH FROM HD.NGAYTT) = 7 AND PDV.TIENDV > 10000000
4  AND CTPDV.MAPDV = PDV.MAPDV AND DV.MADV = CTPDV.MADV AND TENDV = 'Casino' AND KH.MAKH = HD.MAKH
5  AND TENCN = 'CN Huế' AND CN.MACN = KH.MACN AND QUOCTICH = 'Việt Nam';
```

Script Output | Query Result | SQL | All Rows Fetched: 2 in 0.012 seconds

	MAKH	TENKH
1	KH0007	Nguyễn Hoàng Tú
2	KH0009	Hoàng Minh Tuấn

- Tiến hành chạy explain query Q

```
1  SELECT /*+ GATHER_PLAN_STATISTICS */ KH.MAKH, KH.TENKH
2  FROM HOADON HD , PHIEUDICHVU PDV, CT_PHIEUDICHVU CTPDV, DICHVU DV, KHACHHANG KH, CHINHANH CN
3  WHERE HD.MAPDV = PDV.MAPDV AND EXTRACT(MONTH FROM HD.NGAYTT) = 7 AND PDV.TIENDV > 10000000
4  AND CTPDV.MAPDV = PDV.MAPDV AND DV.MADV = CTPDV.MADV AND TENDV = 'Casino' AND KH.MAKH = HD.MAKH
5  AND TENCN = 'CN Huế' AND CN.MACN = KH.MACN AND QUOCTICH = 'Việt Nam';
6
7  SELECT * FROM TABLE(DBMS_XPLAN.display_cursor(format=>'ALLSTATS LAST'));
8
```

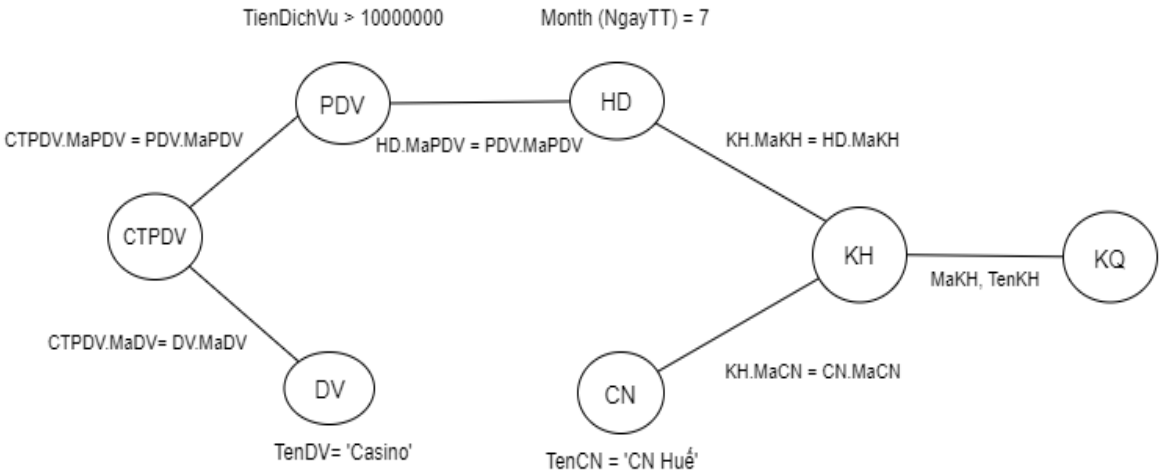
- Ta được kết quả sau

```
PLAN_TABLE_OUTPUT
1  SQL_ID  cnwhqfunm3u9r, child number 1
2  -------------------------------------
3  SELECT /*+ GATHER_PLAN_STATISTICS */ KH.MAKH, KH.TENKH FROM HOADON HD ,
4  PHIEUDICHVU PDV, CT_PHIEUDICHVU CTPDV, DICHVU DV, KHACHHANG KH,
5  CHINHANH CN WHERE HD.MAPDV = PDV.MAPDV AND EXTRACT(MONTH FROM
6  HD.NGAYTT) = 7 AND PDV.TIENDV > 10000000 AND CTPDV.MAPDV = PDV.MAPDV
7  AND DV.MADV = CTPDV.MADV AND TENDV = 'Casino' AND KH.MAKH = HD.MAKH AND
8  TENCN = 'CN Huế' AND CN.MACN = KH.MACN AND QUOCTICH = 'Việt Nam'
9
10 Plan hash value: 519595909
11
12 ------------------------------------------------------------------------------------------------------------------------------------
13 | Id  | Operation                           | Name              | Starts | E-Rows | A-Rows |   A-Time   | Buffers |  OMem |  1Mem | Used-Mem |
14 ------------------------------------------------------------------------------------------------------------------------------------
15 |   0 | SELECT STATEMENT                    |                   |      1 |        |      2 |00:00:00.01 |      25 |       |       |          |
16 |   1 |  NESTED LOOPS                       |                   |      1 |      1 |      2 |00:00:00.01 |      25 |       |       |          |
17 |   2 |   NESTED LOOPS                      |                   |      1 |      1 |      2 |00:00:00.01 |      23 |       |       |          |
18 |*  3 |    HASH JOIN                        |                   |      1 |      1 |      2 |00:00:00.01 |      21 |  1106K|  1106K|  523K (0)|
19 |   4 |     NESTED LOOPS                    |                   |      1 |      1 |      2 |00:00:00.01 |      20 |       |       |          |
20 |   5 |      NESTED LOOPS                   |                   |      1 |      1 |      2 |00:00:00.01 |      16 |       |       |          |
21 |*  6 |       HASH JOIN                     |                   |      1 |      1 |      3 |00:00:00.01 |      12 |  1476K|  1476K| 1075K (0)|
22 |*  7 |        TABLE ACCESS FULL            | HOADON            |      1 |      1 |      5 |00:00:00.01 |       6 |       |       |          |
23 |*  8 |        TABLE ACCESS FULL            | PHIEUDICHVU       |      1 |      1 |     10 |00:00:00.01 |       6 |       |       |          |
24 |*  9 |       TABLE ACCESS BY INDEX ROWID   | KHACHHANG         |      3 |      1 |      2 |00:00:00.01 |       4 |       |       |          |
25 |* 10 |        INDEX UNIQUE SCAN            | PK_KHACHHANG      |      3 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
26 |* 11 |      TABLE ACCESS BY INDEX ROWID    | CHINHANH          |      2 |      1 |      2 |00:00:00.01 |       4 |       |       |          |
27 |* 12 |       INDEX UNIQUE SCAN             | PK_CHINHANH       |      2 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
28 |  13 |     INDEX FULL SCAN                 | PK_CT_PHIEUDICHVU |      1 |      1 |     25 |00:00:00.01 |       1 |       |       |          |
29 |* 14 |    INDEX UNIQUE SCAN                | PK_DICHVU         |      2 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
30 |* 15 |   TABLE ACCESS BY INDEX ROWID       | DICHVU            |      2 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
31 ------------------------------------------------------------------------------------------------------------------------------------
```

```
32
33 Predicate Information (identified by operation id):
34 ---------------------------------------------------
35
36    3 - access("CTPDV"."MAPDV"="PDV"."MAPDV")
37    6 - access("HD"."MAPDV"="PDV"."MAPDV")
38    7 - filter(("HD"."MAPDV" IS NOT NULL AND EXTRACT(MONTH FROM INTERNAL_FUNCTION("HD"."NGAYTT"))=7))
39    8 - filter("PDV"."TIENDV">10000000)
40    9 - filter("QUOCTICH"='Việt Nam')
41   10 - access("KH"."MAKH"="HD"."MAKH")
42   11 - filter("TENCN"='CN Huế')
43   12 - access("CN"."MACN"="KH"."MACN")
44   14 - access("DV"."MADV"="CTPDV"."MADV")
45   15 - filter("TENDV"='Casino')
46
47 Note
48 -----
49    - this is an adaptive plan
50
```

2. Tối ưu hóa câu truy vấn Q

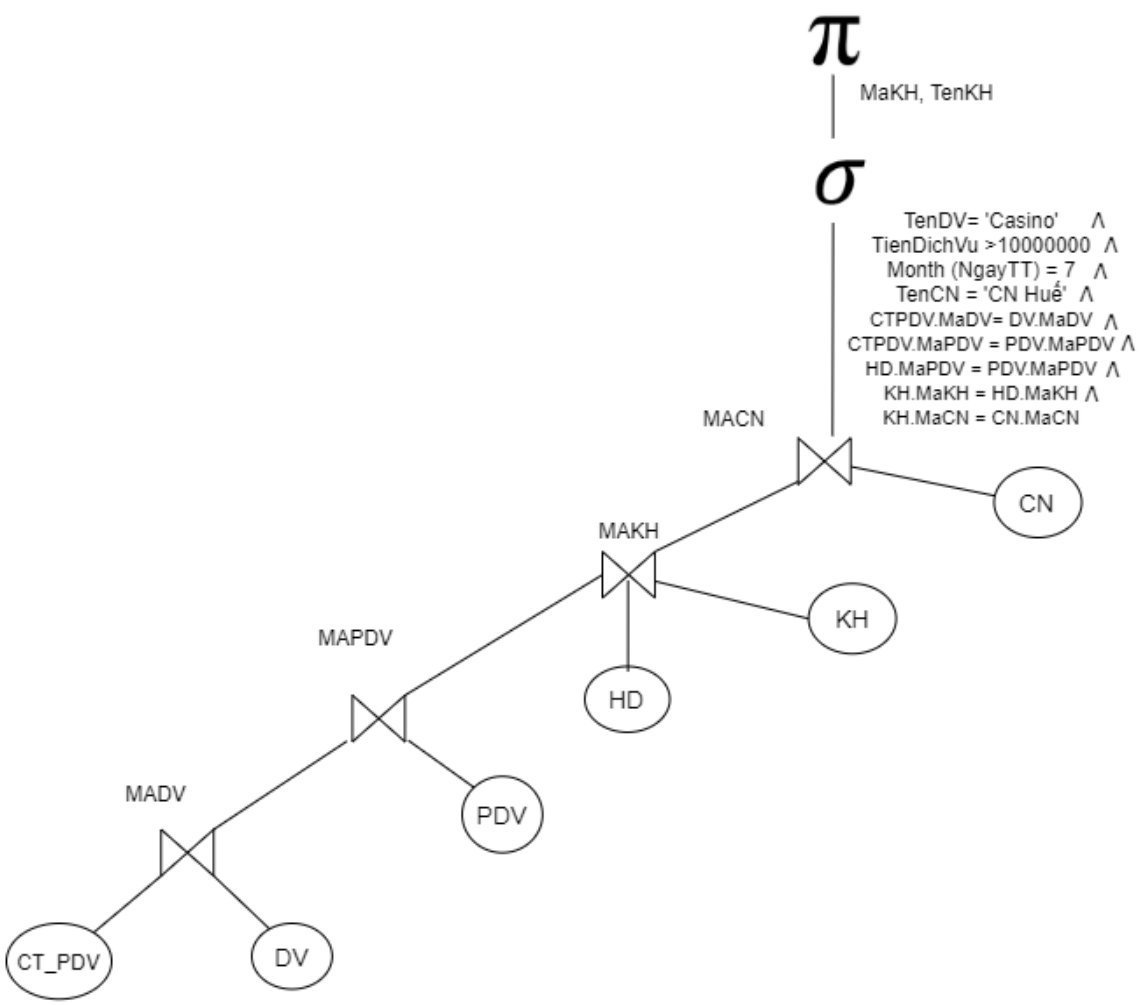
❖ Kiểm tra ngữ nghĩa của truy vấn Q



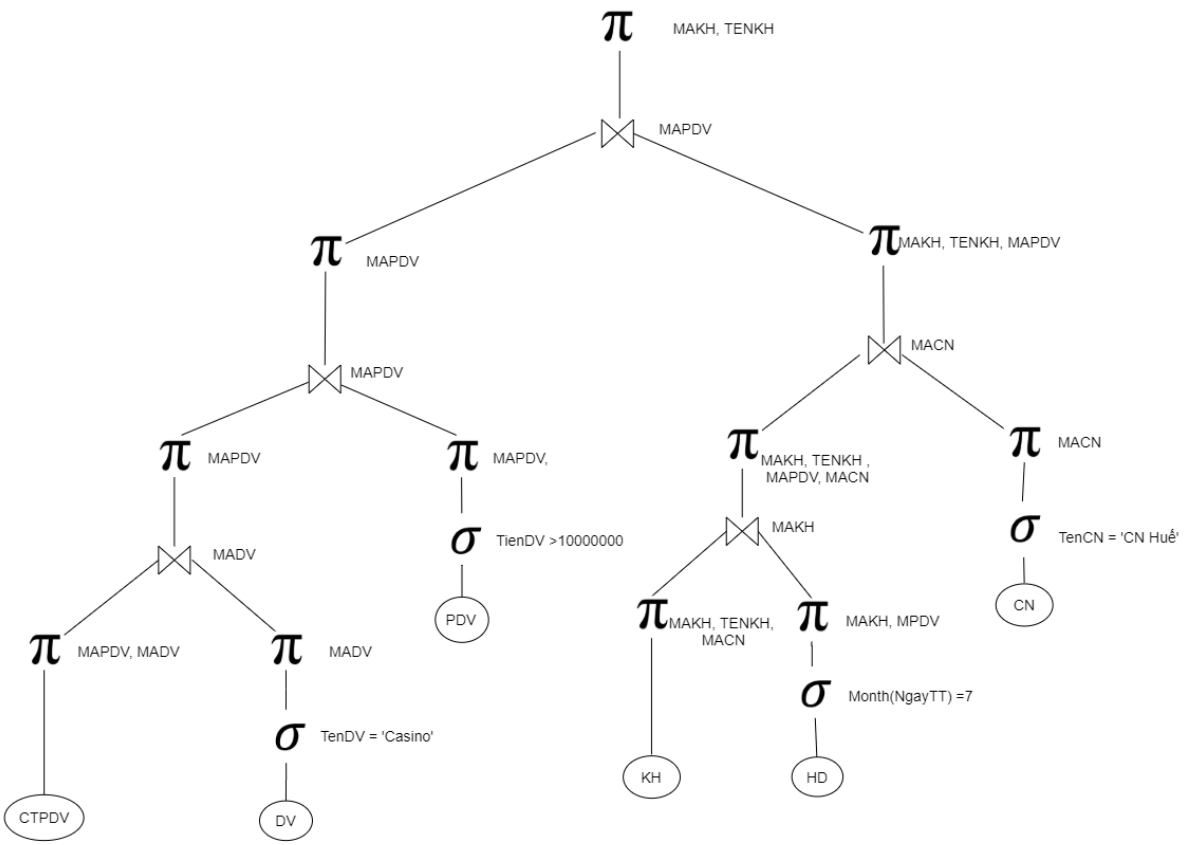
⇨ Đồ thị truy vấn liên thông nên câu truy vấn Q đúng ngữ nghĩa

❖ Phân rã truy vấn để tối ưu hóa toàn cục câu truy vấn Q

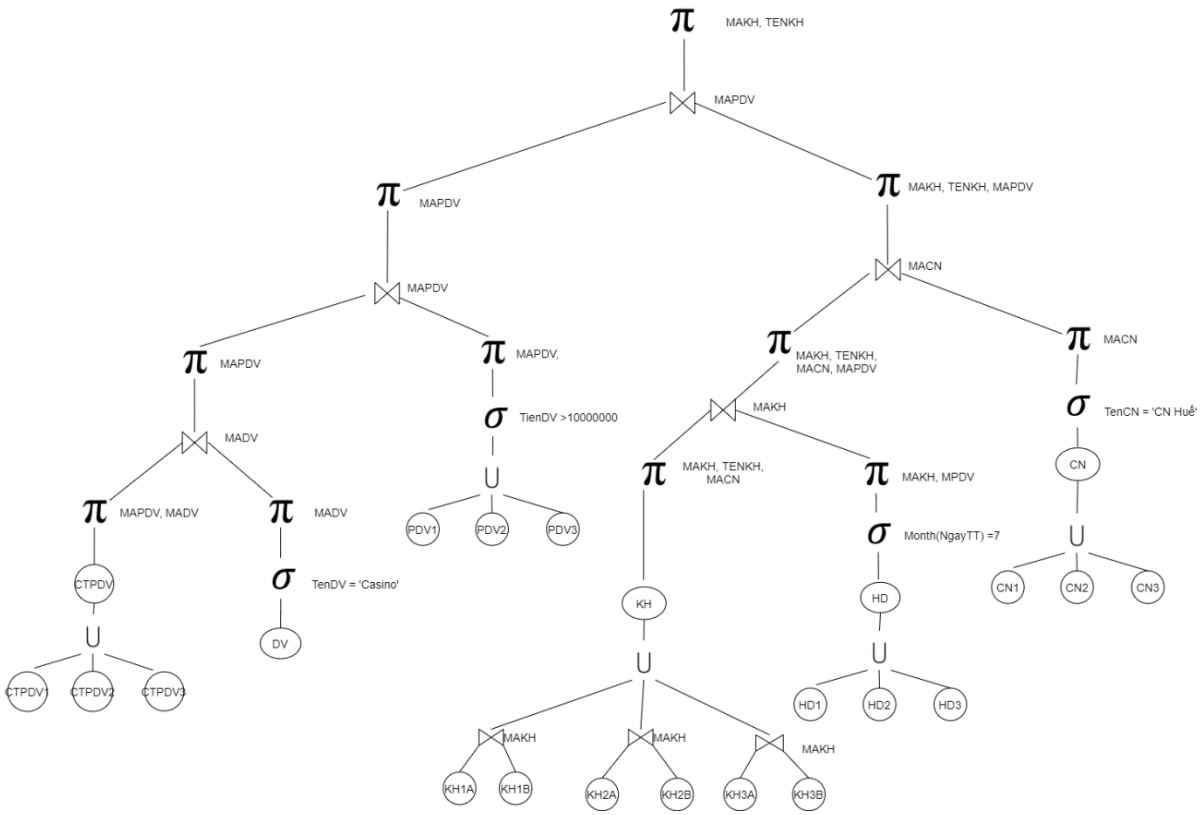
- Cây truy vấn ban đầu:



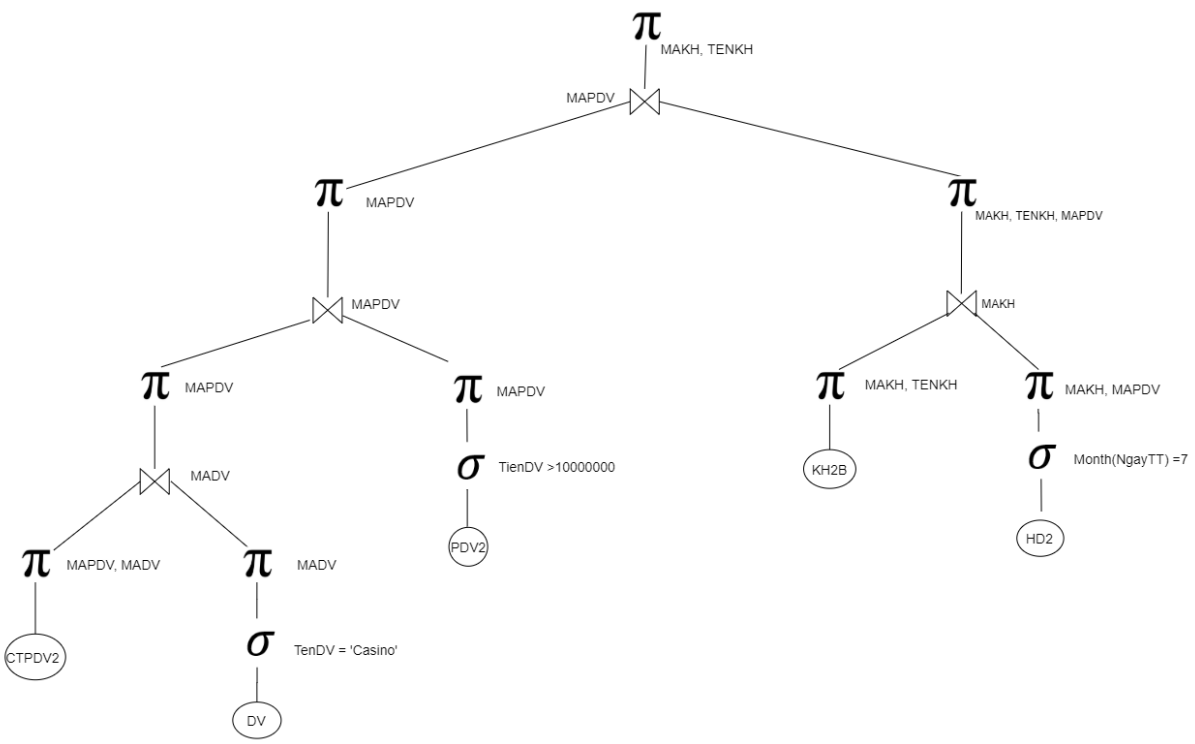
- Cây truy vấn sau khi tối ưu hóa toàn cục



❖ Dựa vào lược đồ phân mảnh ta có cây truy vấn sau :



- Cây truy vấn sau khi được tối ưu hóa



- Câu lệnh SQL của cây truy vấn sau khi được tối ưu hóa

```
SELECT MAKH, TENKH
FROM (  SELECT C.MAPDV
        FROM (  SELECT MAPDV
                FROM (SELECT MAPDV, MADV FROM CT_PHIEUDICHVU)A
JOIN
                     (SELECT MADV FROM DICHVU WHERE TENDV =
'Casino')B
                         ON A.MADV = B.MADV)C
            JOIN
            (   SELECT MAPDV FROM PHIEUDICHVU WHERE TIENDV >
10000000)D
            ON C.MAPDV = D.MAPDV)E
        JOIN
    (   SELECT MAKH, TENKH, MAPDV
        FROM (SELECT MACN FROM CHINHANH WHERE TENCN = 'CN Huế')I
                    JOIN
              ( SELECT G.MAKH, TENKH, MAPDV, MACN
                FROM (SELECT MAKH, TENKH, MACN FROM KHACHHANG)F
                              JOIN
                     (SELECT MAKH, MAPDV FROM HOADON WHERE
EXTRACT(MONTH FROM NGAYTT)= 7 )G
                              ON F.MAKH = G.MAKH)H
```

```
                    ON I.MACN = H.MACN)J
        ON E.MAPDV = J.MAPDV
```

3. Chạy Explain query Q đã được tối ưu

```
57 SELECT /*+ GATHER_PLAN_STATISTICS */ MAKH, TENKH
58 FROM (   SELECT C.MAPDV
59          FROM (   SELECT MAPDV
60                   FROM (SELECT MAPDV, MADV FROM CT_PHIEUDICHVU)A
61                         JOIN
62                        (SELECT MADV FROM DICHVU WHERE TENDV = 'Casino')B
63                         ON A.MADV = B.MADV)C
64              JOIN
65              (    SELECT MAPDV FROM PHIEUDICHVU WHERE TIENDV > 10000000)D
66              ON C.MAPDV = D.MAPDV)E
67          JOIN
68      (    SELECT MAKH, TENKH, MAPDV
69          FROM (SELECT MACN FROM CHINHANH WHERE TENCN = 'CN Huế')I
70                       JOIN
71                 ( SELECT G.MAKH, TENKH, MAPDV, MACN
72                   FROM (SELECT MAKH, TENKH, MACN FROM KHACHHANG)F
73                              JOIN
74                        (SELECT MAKH, MAPDV FROM HOADON WHERE EXTRACT(MONTH FROM NGAYTT)= 7 )G
75                              ON F.MAKH = G.MAKH)H
76                        ON I.MACN = H.MACN)J
77          ON E.MAPDV = J.MAPDV
78
79 SELECT * FROM TABLE(DBMS_XPLAN.display_cursor(format=>'ALLSTATS LAST'));
80
```

- Thu được kết quả như sau

PLAN_TABLE_OUTPUT

```
C.MAPDV          FROM (SELECT MAPDV FROM (SELECT MAPDV, MADV FROM
CT_PHIEUDICHVU)A JOIN (SELECT MADV FROM DICHVU WHERE TENDV = 'Casino')B
ON A.MADV = B.MADV)C              JOIN              (SELECT MAPDV FROM
PHIEUDICHVU WHERE TIENDV > 10000000)D             ON C.MAPDV =
D.MAPDV)E          JOIN      (    SELECT MAKH, TENKH, MAPDV FROM (SELECT
MACN                                        FROM CHINHANH WHERE TENCN
= 'CN Huế')I                                      JOIN
                                ( SELECT G.MAKH, TENKH, MAPDV, MACN
                                    FROM (SELECT MAKH, TENKH, MACN
FROM KHACHHANG)F                                  JOIN (SELECT
MAKH, MAPDV FROM HOADON WHERE EXTRACT(MONTH FROM NGAYTT)= 7 )G ON
F.MAKH = G.MAKH)H         ON I.MACN = H.MACN)J          ON E.MAPDV =
J.MAPDV

Plan hash value: 1395942889

-----------------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name              | Starts | E-Rows | A-Rows |   A-Time     | Buffers |  OMem |  1Mem | Used-Mem |
-----------------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                   |      1 |        |      2 |00:00:00.01 |      29 |       |       |          |
|   1 |  NESTED LOOPS                 |                   |      1 |      1 |      2 |00:00:00.01 |      29 |       |       |          |
|   2 |   NESTED LOOPS                |                   |      1 |      1 |      2 |00:00:00.01 |      27 |       |       |          |
|*  3 |    HASH JOIN                  |                   |      1 |      1 |      3 |00:00:00.01 |      25 |  1061K|  1061K|  462K (0)|
|*  4 |     HASH JOIN                 |                   |      1 |      1 |      4 |00:00:00.01 |      19 |  1106K|  1106K|  779K (0)|
|*  5 |      HASH JOIN                |                   |      1 |      1 |      4 |00:00:00.01 |      18 |  1075K|  1075K|  466K (0)|
|*  6 |       HASH JOIN               |                   |      1 |      1 |      4 |00:00:00.01 |      12 |  1423K|  1423K|  587K (0)|
|*  7 |        TABLE ACCESS FULL      | HOADON            |      1 |      1 |      5 |00:00:00.01 |       6 |       |       |          |
|   8 |        TABLE ACCESS FULL      | KHACHHANG         |      1 |      1 |      5 |00:00:00.01 |       6 |       |       |          |
|*  9 |       TABLE ACCESS FULL       | CHINHANH          |      1 |      1 |      1 |00:00:00.01 |       6 |       |       |          |
|  10 |      INDEX FULL SCAN          | PK_CT_PHIEUDICHVU |      1 |      2 |     25 |00:00:00.01 |       1 |       |       |          |
|* 11 |     TABLE ACCESS FULL         | DICHVU            |      1 |      1 |      1 |00:00:00.01 |       6 |       |       |          |
|* 12 |    INDEX UNIQUE SCAN          | PK_PHIEUDICHVU    |      3 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
|* 13 |   TABLE ACCESS BY INDEX ROWID | PHIEUDICHVU       |      2 |      1 |      2 |00:00:00.01 |       2 |       |       |          |
-----------------------------------------------------------------------------------------------------------------------------------------------
```

```
38
39 Predicate Information (identified by operation id):
40 ---------------------------------------------------
41
42    3 - access("MADV"="MADV")
43    4 - access("MAPDV"="MAPDV")
44    5 - access("MACN"="MACN")
45    6 - access("MAKH"="MAKH")
46    7 - filter(("MAPDV" IS NOT NULL AND EXTRACT(MONTH FROM INTERNAL_FUNCTION("NGAYTT"))=7))
47    9 - filter("TENCN"='CN Huế')
48   11 - filter("TENDV"='Casino')
49   12 - access("MAPDV"="MAPDV")
50   13 - filter("TIENDV">10000000)
51
52 Note
53 -----
54    - this is an adaptive plan
```

- **Nhận xét :** Có thể thấy sau khi được tối ưu hóa query Q tiêu tốn ít dung lượng bộ nhớ CPU (587K so với ban đầu là 1075K) và thời gian thực thi nhanh hơn(1,013s so với ban đầu là 1,015s) và số lượng phân vùng cần được truy cập cũng giảm hơn so với lần ban đầu.

Tài liệu tham khảo

[1]: Oracle Select Statement.

[2]: Oracle Queries

[3]: Create Procedure

[4]: Create function

[5]: Explain Plan

[6]: Database Concept.