

# Bài tập lớn 1: Thiết kế cơ sở dữ liệu phân tán

Nguyễn Khánh Văn  
IS211.O11.HTCL

Trường Đại học Công nghệ Thông tin  
21522781@gm.uit.edu.vn

Lê Thị Kiều Lam  
IS211.O11.HTCL

Trường Đại học Công nghệ Thông tin  
21522275@gm.uit.edu.vn

Đặng Quang Nhật  
IS211.O11.HTCL

Trường Đại học Công nghệ Thông tin  
21522413@gm.uit.edu.vn

Hoàng Quốc Việt  
IS211.O11.HTCL

Trường Đại học Công nghệ Thông tin  
21522790@gm.uit.edu.vn

**Tóm tắt** — Trong bài tập lớn 1 này, nhóm chúng tôi sẽ tiến hành thiết kế một cơ sở dữ liệu phân tán trên Oracle. Sau đó sẽ tiến hành thêm dữ liệu vào, tạo ra các câu truy vấn phức tạp trên môi trường phân tán, giả lập các mức cô lập trong môi trường phân tán, tạo các procedure, trigger, tối ưu hóa một câu truy vấn phức tạp.

**Từ khóa** — Cơ sở dữ liệu phân tán, Oracle, Thiết kế cơ sở dữ liệu phân tán

## I. GIỚI THIỆU

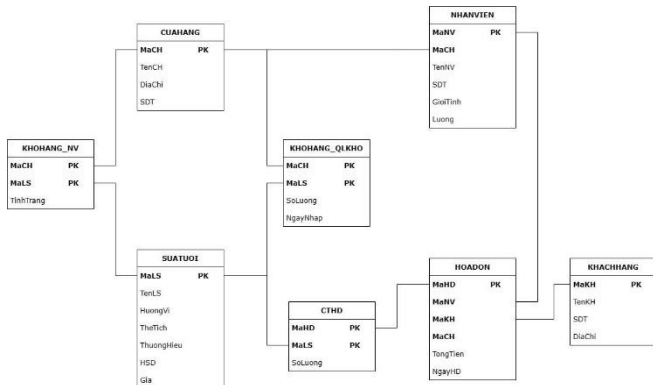
Cơ sở dữ liệu phân tán là một cơ sở dữ liệu được lưu trữ trên nhiều máy tính khác nhau trong một mạng. Điều này mang lại nhiều lợi thế, bao gồm khả năng mở rộng, khả năng sẵn sàng và hiệu suất cao hơn. Cơ sở dữ liệu phân tán có thể được sử dụng trong nhiều ứng dụng khác nhau, chẳng hạn như hệ thống thương mại điện tử, hệ thống quản lý kho hàng và hệ thống quản lý tài chính.

Oracle là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được sử dụng bởi các doanh nghiệp lớn và nhỏ trên khắp thế giới. Oracle có nhiều ưu điểm, bao gồm khả năng mở rộng, khả năng sẵn sàng và hiệu suất cao. Oracle có thể được sử dụng trong nhiều ứng dụng khác nhau, bao gồm hệ thống thương mại điện tử, hệ thống quản lý kho hàng và hệ thống quản lý tài chính.

Trong nghiên cứu này, nhóm chúng tôi sẽ tiến hành thiết kế cơ sở dữ liệu phân tán trên hệ quản trị cơ sở dữ liệu là Oracle để tiến hành thiết kế, tạo ra các câu truy vấn phức tạp, giả lập mức cô lập trên Oracle và tối ưu hóa truy vấn đơn giản.

## II. TÀI NGUYÊN

### A. Sơ đồ diagram cơ sở dữ liệu



### B. Thông tin các bảng

SUATUOI(MaLS, TenLS, HuongVi, TheTich, ThuongHieu, HSD, Gia)

**Tên từ:** Mỗi loại Sữa có mã loại sữa (MaLS) là duy nhất, tên loại sữa (TenLS), hương vị (HuongVi), thể tích của loại sữa (TheTich), thương hiệu (ThuongHieu), Hạn sử dụng (HSD), Giá (Gia).

**CUAHANG**(MaCH, TenCH, DiaChi, SĐT)

**Tên từ:** Mỗi cửa hàng có mã cửa hàng (MaCH) là duy nhất, tên cửa hàng (TenCH), địa chỉ (DiaChi) và số điện thoại hotline của cửa hàng (SĐT).

**NHANVIEN**(MaNV, MaCH, TenNV, SĐT, GioiTinh, Luong)

**Tên từ:** Mỗi nhân viên có mã nhân viên (MaNV) là duy nhất, mã cửa hàng (MaCH) mà nhân viên làm việc, tên nhân viên (TenNV), giới tính nhân viên (GioiTinh), lương của nhân viên đó (Luong).

**KHACHHANG**(MaKH, TenKH, SĐT, DiaChi)

**Tên từ:** Mỗi khách hàng có mã khách hàng (MaKH) là duy nhất, tên khách hàng (TenKH), số điện thoại mua hàng (SĐT), địa chỉ (DiaChi).

**KHOHANG\_QLKHO**(MaCH, MaLS, SoLuong, NgayNhap)

**Tên từ:** Mỗi loại sữa khi được nhập vào kho hàng để quản lý sẽ lưu lại thông tin mã cửa hàng (MaCH), mã loại sữa đã nhập (MaLS), số lượng nhập (SoLuong), ngày nhập (NgayNhap).

**KHOHANG\_NV**(MaCH, MaLS, TinhTrang)

**Tên từ:** Mỗi loại sữa khi được nhân viên bán ra sẽ được mô tả qua các thuộc tính mã cửa hàng của nhân viên (MaCH), mã loại sữa (MaLS), tình trạng (TinhTrang) còn hàng hoặc hết hàng.

**HOADON**(MaHD, MaNV, MaKH, MaCH, TongTien, NgayHD)

**Tên từ:** Mỗi hóa đơn thanh toán sẽ lưu lại thông tin mã hóa đơn (MaHD) là duy nhất, mã nhân viên bán (MaNV), mã khách hàng đã thanh toán (MaKH), mã cửa hàng (MaCH), tổng tiền của hóa đơn (TongTien), ngày thanh toán (NgayHD).

**CTHD**(MaHD, MaLS, SoLuong)

**Tên từ:** Mỗi hóa đơn sẽ lưu giữ thông tin chi tiết của hóa đơn bao gồm mã hóa đơn (MaHD), mã loại sữa (MaLS), số lượng bán ra (SoLuong).

### C. Kiến trúc phân quyền

User	Role
------	------

GiamDoc	CONNECT, CREATE DATABASELINK
QuanLy	CONNECT
NhanVien	CONNECT

#### Phân quyền:

- Cửa hàng 1:
  - GiamDoc: Xem được toàn bộ thông tin của cả hai chi nhánh.
  - QuanLy:
    - + Xem được thông tin KHACHHANG, SUATUOI, HOADON, CTHD của cả 2 chi nhánh.
    - + Xem được thông tin KHOHANG\_QLKHO, KHOHANG\_NV, NHANVIEN của chi nhánh đang quản lí.
    - NhanVien: Xem được thông tin KHOHANG\_NV, SUATUOI, HOADON, CTHD, KHACHHANG của chi nhánh đang làm việc.
- Cửa hàng 2:
  - QuanLy:
    - + Xem được thông tin KHACHHANG, SUATUOI, HOADON, CTHD của cả 2 chi nhánh.
    - + Xem được thông tin KHOHANG\_QLKHO, KHOHANG\_NV, NHANVIEN của chi nhánh đang quản lí.
    - NhanVien: Xem được thông tin KHOHANG\_NV, SUATUOI, HOADON, CTHD, KHACHHANG của chi nhánh đang làm việc.

### III. THỰC HIỆN TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

#### A. Liệt kê (MaNV, TenNV, SDT) ở cả 2 cửa hàng (hội) có lương trên 5000000

```
SELECT MaNV, TenNV, SDT
FROM CH1.NHANVIEN WHERE Luong >= 5000000
UNION
SELECT MaNV, TenNV, SDT
FROM CH2.NHANVIEN@giamdoc1_dbblink
WHERE Luong >= 5000000
```

Kết quả:

	MANV	TENNV	SDT
1	NV001	Nguyễn Khánh Văn	0123456789
2	NV007	Ngô Văn Mạnh	0869574321
3	NV021	Nguyễn Ngọc Lam	0123456789
4	NV027	Đoàn Khắc Huy	0869574321

#### B. Tìm thông tin sữa (MaLS, TenLS, Gia) HươngVi được phân phối đến cả 2 cửa hàng với tình trạng còn hàng (Giao).

```
SELECT ST1.MaLS, TenLS, HươngVi, Gia
FROM CH1.SUATUOI ST1 JOIN
CH1.KHOHANG_NV NV1 ON ST1.MaLS =
NV1.MaLS
WHERE TìnhTrang = 'Còn hàng'
INTERSECT
SELECT ST2.MaLS, TenLS, HươngVi, Gia
FROM CH2.SUATUOI@nv1_dbblink ST2 JOIN
CH2.KHOHANG_NV@nv1_dbblink NV2 ON
NV2.MaLS = NV2.MaLS
WHERE TìnhTrang = 'Còn hàng'
```

Kết quả:

	MALs	TENLS	HƯƠNGVI	GIA
1	LS090	Sữa tươi tiệt trùng DaLat Milk Ngọt		35000

#### C. Tìm doanh thu trong tháng 12 ở mỗi cửa hàng

```
SELECT HD1.MaCH, SUM(TongTien) AS DOANHTHU
FROM CH1.HOADON HD1
WHERE To_char (NgàyHD, 'mm') = '12' GROUP BY
HD1.MaCH
UNION
SELECT HD2.MaCH, SUM(TongTien) DOANHTHU
FROM CH2.HOADON@giamdoc1_dbblink HD2
WHERE To_char (NgàyHD, 'mm') = '12' GROUP BY
HD2.MaCH
```

Kết quả:

	MaCH	DOANHTHU
1	CH001	1410000
2	CH002	2410000

#### D. Tìm thông tin sữa gồm mã Loại sữa, tên loại sữa, tổng số lượng bán được của sữa bán chạy nhất tại cửa hàng quản lý.

```
SELECT ST.MaLS, TenLS, SUM(SoLuong)
FROM CH1.SUATUOI ST, CH1.CTHD CT
WHERE CT.MaLS = ST.MaLS
GROUP BY ST.MaLS, TenLS
HAVING SUM(SoLuong) >= ALL
(SELECT SUM(SoLuong) FROM CH1.CTHD
GROUP BY MaLS)
```

Kết quả:

```
--Cau4
SELECT ST.MaLS, TenLS, SUM(SoLuong)
FROM CH1.SUATUOI ST, CH1.CTHD CT
WHERE CT.MaLS = ST.MaLS
GROUP BY ST.MaLS, TenLS
HAVING SUM(SoLuong) >= ALL
(SELECT SUM(SoLuong) FROM CH1.CTHD GROUP BY MaLS)
```

MaLS	TenLS	SUM(SOLUONG)
1 LS004	Sữa tươi tiết trùng Vinamilk	32

E. Tìm những khách hàng chỉ mua sữa ở CH1 (trừ).

```
SELECT DISTINCT K.MaKH, TenKH
FROM CH1.KHACHHANG K1, CH1.HOADON H1
WHERE K1.MaKH = H1.MaKH
MINUS
SELECT DISTINCT K.MaKH, TenKH
FROM CH2.KHACHHANG K2,
CH2.HOADON@giamdoc1_dbblink H2
WHERE K2.MaKH = H2.MaKH
```

Kết quả:

```
--Cau5
SELECT DISTINCT K1.MaKH, TenKH
FROM CH1.KHACHHANG K1, CH1.HOADON H1
WHERE K1.MaKH = H1.MaKH
MINUS
SELECT DISTINCT K2.MaKH, TenKH
FROM CH2.KHACHHANG@giamdoc1_dbblink K2, CH2.HOADON@giamdoc1_dbblink H2
WHERE K2.MaKH = H2.MaKH
```

MaKH	TenKH
1 KH001	Nguyễn Văn A
2 KH002	Trần Thị B
3 KH003	Nguyễn Văn C
4 KH004	Trần Thị D
5 KH005	Lê Văn E
6 KH006	Phạm Thị F
7 KH007	Võ Văn G
8 KH008	Nguyễn Thị H
9 KH009	Trần Văn I
10 KH010	Lê Thị K

F. Tìm khách hàng (MaKH) đã mua tất cả sữa của thương hiệu TH True Milk ở cửa hàng quản lý (chia).

```
SELECT MaKH
FROM CH1.HOADON HD
WHERE NOT EXISTS
(SELECT * FROM CH1.SUATUOI ST
WHERE THUONGHIEU = 'TH True Milk' AND NOT EXISTS
(SELECT * FROM CH1.CTHD CT
WHERE CT.MaHD=HD.MaHD AND CT.MaLS =ST.MaLS))
```

Kết quả:

```
--Cau6
SELECT MaKH
FROM CH1.HOADON HD
WHERE NOT EXISTS
(SELECT * FROM CH1.SUATUOI ST
WHERE THUONGHIEU = 'TH True Milk' AND NOT EXISTS
(SELECT * FROM CH1.CTHD CT
WHERE CT.MaHD=HD.MaHD AND CT.MaLS =ST.MaLS))
```

MaKH
1 KH001

G. Tìm khách hàng đã mua sữa ở cả 2 cửa hàng (giao).

```
SELECT K1.MaKH, TenKH
FROM CH1.KHACHHANG K1, CH1.HOADON H1
WHERE K1.MaKH = H1.MaKH
INTERSECT
SELECT K2.MaKH, TenKH
FROM CH2.KHACHHANG@giamdoc1_dbblink K2,
CH2.HOADON@giamdoc1_dbblink H2
WHERE K2.MaKH = H2.MaKH
```

Kết quả:

```
--Cau 7
SELECT K1.MaKH, TenKH
FROM CH1.KHACHHANG K1, CH1.HOADON H1
WHERE K1.MaKH = H1.MaKH
INTERSECT
SELECT K2.MaKH, TenKH
FROM CH2.KHACHHANG@giamdoc1_dbblink K2, CH2.HOADON@giamdoc1_dbblink H2
WHERE K2.MaKH = H2.MaKH
```

MaKH	TenKH
1 KH090	Tà Thị Nua

H. Đưa ra thông tin 3 loại sữa bán chạy nhất trong tháng 12 theo từng cửa hàng. MaCH, MaLS, TenLS.

```
SELECT *
FROM (SELECT CT1.MaLS, TenLS, ThuongHieu,
MACH, SUM(SOLUONG)
FROM CH1.SUATUOI ST1, CH1.HOADON HD1,
CH1.CTHD CT1
WHERE ST1.MaLS = CT1.MaLS AND CT1.MaHD =
HD1.MaHD AND To_char (NgaHD,'mm') = '12'
GROUP BY CT1.MaLS, TenLS, ThuongHieu, MaCH
ORDER BY SUM(SoLuong) DESC)
WHERE Rownum <= 3
UNION
SELECT *
FROM (SELECT CT2.MaLS, TenLS,
ThuongHieu, MaCH, SUM(SoLuong)
FROM CH2.SUATUOI@quanly1_dbblink ST2,
CH2.HOADON@quanly1_dbblink HD2,
CH2.CTHD@quanly1_dbblink CT1
WHERE ST2.MaLS=CT2.MaLS AND
CT2.MaHD=HD2.MaHD AND To_char
(NgayHD,'mm') = '12'
GROUP BY CT2.MaLS, TenLS, ThuongHieu, MaCH
ORDER BY SUM(SoLuong) DESC)
WHERE Rownum <= 3
```

Kết quả:

```
--Cau8
SELECT *
FROM (SELECT CT1.MaLS, TenLS, ThuongHieu, MaCH, SUM(SOLUONG)
FROM CH1.SUATUOI ST1, CH1.HOADON HD1,
CH1.CTHD CT1
WHERE ST1.MaLS = CT1.MaLS AND CT1.MaHD = HD1.MaHD AND To_char (NgaHD,'mm') = '12'
GROUP BY CT1.MaLS, TenLS, ThuongHieu, MaCH
ORDER BY SUM(SoLuong) DESC)
WHERE Rownum <= 3
UNION
SELECT *
FROM (SELECT CT2.MaLS, TenLS, ThuongHieu, MaCH, SUM(SOLUONG)
FROM CH2.SUATUOI@quanly1_dbblink ST2, CH2.HOADON@quanly1_dbblink HD2,
CH2.CTHD@quanly1_dbblink CT2
WHERE ST2.MaLS=CT2.MaLS AND CT2.MaHD=HD2.MaHD AND To_char (NgaHD,'mm') = '12'
GROUP BY CT2.MaLS, TenLS, ThuongHieu, MaCH
ORDER BY SUM(SoLuong) DESC)
WHERE Rownum <= 3
```

MaLS	TenLS	THUONGHIEU	MaCH	SUM(SOLUONG)
1 LS001	Sữa tươi tiết trùng Vinamilk	Vinamilk	CH001	34
2 LS003	Sữa tươi tiết trùng Vinamilk	Vinamilk	CH001	26
3 LS004	Sữa tươi tiết trùng Vinamilk	Vinamilk	CH001	32
4 LS001	Sữa tươi tiết trùng Vinamilk	Vinamilk	CH001	32
5 LS004	Sữa tươi tiết trùng Vinamilk	Vinamilk	CH002	24
6 LS009	Sữa tươi nguyên chất thanh trùng TH True Milk	TH True Milk	CH002	60

I. In ra danh sách các loại sữa (MaLS, TenLS) không bán được ở tất cả các cửa hàng.

```
SELECT MaLS, TenLS
```

```

FROM CH1.SUATUOI
WHERE MaLS NOT IN (SELECT MaLS FROM
CH1.CTHD)
INTERSECT
SELECT MaLS, TenLS FROM
CH2.SUATUOI@giamdoc1_dblink
WHERE MaLS NOT IN (SELECT MaLS FROM
CH2.CTHD@giamdoc1_dblink)

```

Kết quả:

MaLS	TenLS
1 LS090	Sữa tươi tiệt trùng DaLat Milk

J. Tìm những loại sữa mà CH1 chưa có (tình trạng còn hàng) mà CH2 lại có (tình trạng hết hàng).

```

SELECT ST1.MaLS, TenLS
FROM CH1.SUATUOI ST1, CH1.KHOHANG_NV
KH1
WHERE ST1.MaLS = KH1.MaLS AND
KH1.TinhTrang = 'Còn hàng'
UNION
SELECT ST2.MaLS, TenLS
FROM CH2.SUATUOI@giamdoc1_dblink ST2,
CH2.KHOHANG_NV@giamdoc1_dblink KH2
WHERE ST2.MaLS = KH2.MaLS AND
KH2.TinhTrang = 'Hết hàng'

```

Kết quả:

MaLS	TenLS
1 LS001	Sữa tươi tiệt trùng Vinamilk
2 LS002	Sữa tươi tiệt trùng Vinamilk
3 LS004	Sữa tươi tiệt trùng Vinamilk
4 LS006	Sữa tươi nguyên chất thanh trùng TH True Milk
5 LS009	Sữa tươi nguyên chất thanh trùng TH True Milk
6 LS010	Sữa tươi nguyên chất thanh trùng TH True Milk
7 LS021	Sữa tươi tiệt trùng Vinamilk
8 LS027	Sữa tươi nguyên chất thanh trùng TH True Milk
9 LS030	Sữa tươi nguyên chất thanh trùng TH True Milk
10 LS090	Sữa tươi tiệt trùng DaLat Milk

#### IV. FUNCTION, PROCEDURE, TRIGGER TRÊN MÔI TRƯỜNG PHẦN TẤN

##### A. Function

Function	Tính tổng tiền tất cả các hóa đơn khách hàng chỉ trả	
Function Name	sumPrice	
Arguments	MaKH	

Output	Tổng tiền các hóa đơn của khách hàng có mã khách hàng là MaKH
Query	<pre> CREATE OR REPLACE FUNCTION sumPrice (customerId KHACHHANG.MaKH%TYPE) RETURN NUMBER AS totalPrice NUMBER; BEGIN SELECT SUM(TOTAL) INTO totalPrice FROM ( SELECT SUM(TongTien) AS TOTAL FROM HOADON WHERE MaKH=customerId UNION SELECT SUM (TongTien) AS TOTAL FROM HOADON@giamdoc1_dblink WHERE MaKH=customerId); RETURN TongTien; END; </pre>
Execute Procedure	<pre> SET SERVEROUTPUT ON; DECLARE t NUMBER; BEGIN t := sumPrice('KH009'); DBMS_OUTPUT.PUT_LINE('Tong tien: '   t); END; </pre>

##### B. Trigger

- Ràng buộc toàn vẹn:  
+ Bối cảnh: SUATUOI
- Nội dung:  $\forall t \in \text{SUATUOI}: t.GIA > 0$
- Bảng tầm ảnh hưởng:

R1	Thêm	Xóa	Sửa
SUATUOI	+	-	+(GIA)

```

CREATE OR REPLACE TRIGGER
TR_CHECK_GIA_INSERT
BEFORE INSERT OR UPDATE ON SUATUOI FOR
EACH ROW
BEGIN
IF (:NEW.Gia <= 0)
THEN
RAISE_APPLICATION_ERROR(-20100, 'Gia xe khong
hop le, xin vui long nhap lai');
END IF;
END;

```

##### C. Procedure

Procedure	Tính tổng tiền tất cả các hóa đơn khách hàng chỉ trả
-----------	--

<b>Procedure Name</b>	PROCEDURE1	
<b>Arguments</b>	MaKH	Mã khách hàng
<b>Output</b>	Số lượng sản phẩm mà khách hàng mua nhiều nhất ở từng cửa hàng	
<b>Query</b>	<b>CREATE OR REPLACE PROCEDURE</b> <b>PROCEDURE1(CUSID</b> <b>KHACHHANG.MAKH%TYPE)</b> <b>AS</b> <b>BEGIN</b> <b>FOR r IN (</b> <b>SELECT HD.MaCH,</b> <b>CTHD1.MaLS, ST.TenLS,</b> <b>SUM(CTHD1.SoLuong) AS</b> <b>TongSoLuong</b> <b>FROM CH1.CTHD CTHD1</b> <b>JOIN CH1.SUATUOI ST ON</b> <b>CTHD1.MaLS = ST.MaLS</b> <b>JOIN CH1.HOADON HD</b> <b>ON HD.MAHD = CTHD1.MAHD</b> <b>WHERE HD.MaKH =</b> <b>CUSID</b> <b>GROUP BY HD.MaCH,</b> <b>CTHD1.MaLS, ST.TenLS</b> <b>UNION ALL</b> <b>SELECT HD.MaCH,</b> <b>CTHD2.MaLS, ST.TenLS,</b> <b>SUM(CTHD2.SoLuong) AS</b> <b>TongSoLuong</b> <b>FROM</b> <b>CH2.CTHD@giamdoc1_dblink</b> <b>CTHD2</b> <b>JOIN</b> <b>CH2.SUATUOI@giamdoc1_dblink</b> <b>ST ON CTHD2.MaLS = ST.MaLS</b> <b>JOIN</b> <b>CH2.HOADON@giamdoc1_dblink</b> <b>HD ON HD.MAHD =</b> <b>CTHD2.MAHD</b> <b>WHERE HD.MaKH =</b> <b>CUSID</b> <b>GROUP BY HD.MaCH,</b> <b>CTHD2.MaLS, ST.TenLS</b> <b>)</b> <b>LOOP</b> DBMS_OUTPUT.PUT_LINE('Cua hang '    r.MaCH    ': San pham '    r.MaLS    ' ('    r.TenLS    ') - Tong so luong: '    r.TongSoLuong); <b>END LOOP;</b> <b>END;</b>	
<b>Execute Procedure</b>	<b>SET SERVEROUTPUT ON</b> <b>BEGIN</b> <b>PROCEDURE1('KH090');</b>	

	<b>END;</b>
--	-------------

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

### A. Lost update

Nhân viên 1 đang thay đổi thông tin của khách hàng thì có nhân viên 2 đến thay đổi thông tin cũng của chính khách hàng đó nhưng với dữ liệu khác. Từ đó, dẫn đến việc thông tin của nhân viên 2 ghi đè lên trên thông tin của nhân viên 1.

Thời gian	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 10' <b>WHERE</b> MaKH=KH021;	
<b>T1</b>	<b>COMMIT;</b>	
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 11' <b>WHERE</b> MaKH=KH021;
<b>T3</b>		<b>COMMIT;</b>
<b>T4</b>	<b>SELECT * FROM</b> CH2.KHACHHANG;	
<b>Result</b>	Quận 11 DiaChi đã bị ghi đè	

Cách ngăn chặn:

Thời gian	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;
<b>T1</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 14' <b>WHERE</b> MaKH='KH021';	
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 11' <b>WHERE</b> MaKH='KH021';
<b>T3</b>	<b>COMMIT;</b>	
<b>T4</b>		Error report - ORA-08177: can't serialize access for this transaction ORA-02063: preceding line from GIAMDOC1_DBLINK

<b>T5</b>	<b>ELECT</b> DiaChi <b>FROM</b> CH2.KHACHHANG <b>WHERE</b> MaKH ='KH021'	
<b>Res ult</b>	Quận 14	

#### B. Non-repeatable

Nhân viên 1 xem thông tin khách hàng lần 1 hoàn tất thì nhân viên 2 truy cập vào hệ thống để thay đổi thông tin khách hàng. Sau đó, nhân viên 1 quay lại để kiểm tra thông tin thì nhận thấy có sự thay đổi so với lần xem đầu tiên.

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>SELECT</b> DiaChi <b>FROM</b> CH2.KHACHHANG <b>WHERE</b> MaKH ='KH021'	
<b>T1</b>		
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG @giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 1' <b>WHERE</b> MaKH='KH021';
<b>T3</b>		<b>COMMIT;</b>
<b>T4</b>	<b>SELECT</b> DiaChi <b>FROM</b> CH2.KHACHHANG <b>WHERE</b> MaKH='KH021'	
<b>Res ult</b>	Quận 1 Khác với dữ liệu truy vấn ban đầu	

Cách ngăn chặn:

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;
<b>T1</b>	<b>SELECT</b> DiaChi <b>FROM</b> CH2.KHACHHANG <b>WHERE</b> MaKH ='KH021' ----- Kết quả : Quận 9	
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 1' <b>WHERE</b> MaKH= 'KH021'
<b>T3</b>		<b>COMMIT;</b>
<b>T4</b>	<b>SELECT * FROM</b> CH2.KHACHHANG;	
<b>Res ult</b>	Quận 9	

→ Dữ liệu vẫn giữ nguyên và chỉ nhận thay đổi khi cửa hàng 2 thực hiện commit mới thành 'Quận 1'

#### C. Deadlock

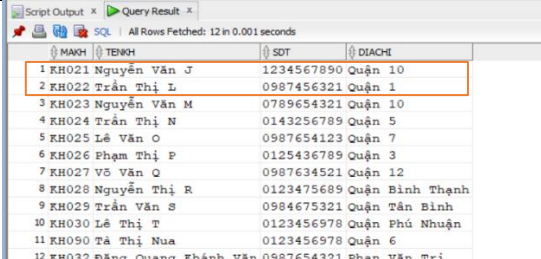
Nhân viên 1 thay đổi thông tin của khách hàng KH021 trong lúc đó nhân viên 2 thay đổi thông tin của khách hàng KH022. Sau đó, nhân viên 1 chuyển sang thay đổi thông tin của khách hàng KH022 còn nhân viên 2 cũng làm ngược lại. Kết quả sẽ bị circular deadlock.

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = READ COMMITTED;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = READ COMMITTED;
<b>T1</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 10' <b>WHERE</b> MaKH='KH021';	
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 8' <b>WHERE</b> MaKH='KH022';
<b>T3</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 9' <b>WHERE</b> MaKH='KH022';	
<b>T4</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 6' <b>WHERE</b> MaKH='KH021';
<b>Res ult</b>	Deadlock	

Cách ngăn chặn:

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;
<b>T1</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 10' <b>WHERE</b> MaKH='KH021';	
<b>T2</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dblink <b>SET</b> DiaChi = 'Quận 1' <b>WHERE</b> MaKH ='KH022';



<b>T3</b>	<b>UPDATE</b> CH2.KHACHHANG <b>SET</b> DiaChi = 'Quận 8' <b>WHERE</b> MaKH ='KH022';	
<b>T4</b>		<b>UPDATE</b> CH2.KHACHHANG@ giamdoc1_dbblink <b>SET</b> DiaChi = 'Quận 2' <b>WHERE</b> MaKH ='KH021';
<b>T5</b>	ORA-00060: deadlock detected while waiting for resource	Error report - ORA-02049: timeout: distributed transaction waiting for lock
<b>T6</b>	<b>COMMIT;</b>	
<b>T7</b>		<b>COMMIT;</b>
<b>Res ult</b>		

#### D. Dirty read

Trong hệ quản trị cơ sở dữ liệu Oracle, mức cô lập mặc định là read committed vì vậy Dirty Read không được cho phép.

#### E. Phantom read

Nhân viên 1 xem thông tin khách hàng lần 1 hoàn tất thì nhân viên 2 truy cập vào hệ thống để thêm thông tin khách hàng mới. Sau đó, nhân viên 1 quay lại để kiểm tra thông tin thì nhận thấy có sự thay đổi so với lần xem đầu tiên.

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = READ COMMITTED;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = READ COMMITTED;
<b>T1</b>	<b>SELECT * FROM</b> CH2.KHACHHANG; ----- Hiện tại dữ liệu vẫn giữ nguyên, chưa có dữ liệu mà bên CH2 thêm vào.	
<b>T2</b>		<b>INSERT INTO</b> CH2.KHACHHANG@ giamdoc1_dbblink <b>VALUES</b> ('KH032','Đặng Quang Vân','142356789','Quận Gò Vấp');
<b>T3</b>		<b>COMMIT;</b>
<b>T4</b>	<b>SELECT * FROM</b> CH2.KHACHHANG;	
<b>Res ult</b>		

8 KH028 Nguyễn Thị R	0123475689 Quận Bình Thạnh
9 KH029 Trần Văn S	0984675321 Quận Tân Bình
10 KH030 Lê Thị T	0123456978 Quận Phú Nhuận
11 KH032 Đặng Quang Văn	142356789 Quận Gò Vấp

Cách ngăn chặn:

Time	Cửa hàng 2	Cửa hàng 1
<b>T0</b>	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;	<b>ALTER SESSION</b> <b>SET</b> ISOLATION_LEVEL = SERIALIZABLE;
<b>T1</b>	<b>SELECT * FROM</b> CH2.KHACHHANG; ----- Dữ liệu vẫn như cũ chưa có gì thay đổi	
<b>T2</b>		<b>INSERT INTO</b> CH2.KHACHHANG@ giamdoc1_dbblink <b>VALUES</b> ('KH032','Đặng Quang Vân','142356789','Quận Gò Vấp');
<b>T3</b>		<b>COMMIT;</b>
<b>T4</b>	<b>SELECT * FROM</b> CH2.KHACHHANG;	
<b>Res ult</b>	Dữ liệu vẫn giữ nguyên khi Cửa hàng 2 xem thông tin khách hàng. →Dữ liệu chỉ nhận thay đổi khi Cửa hàng 2 commit.	

## VI. THỰC HIỆN TỐI ƯU HÓA TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN 1 CÂU TRUY VẤN ĐƠN GIẢN

### A. Lược đồ phân mảnh

- Quan hệ CUAHANG phân mảnh ngang chính theo “DiaChi”:

CUAHANG1 =  $\sigma_{\text{DiaChi} = 'Q1 \text{ Tp.HCM}'}$  (CUAHANG)  
CUAHANG2 =  $\sigma_{\text{DiaChi} = 'Q7 \text{ Tp.HCM}'}$  (CUAHANG)

- Quan hệ NHANVIEN, KHOHANG\_QLKHO, KHOHANG\_NV, HOADON, CTHD phân mảnh ngang dẫn xuất như sau:

NHANVIEN1 = NHANVIEN  $\bowtie$  MaCH (CUAHANG1)  
NHANVIEN2 = NHANVIEN  $\bowtie$  MaCH (CUAHANG2)  
KHOHANG\_QLKHO1 = KHOHANG\_QLKHO  $\bowtie$  MaCH (CUAHANG1)

KHOHANG\_QLKHO2 = KHOHANG\_QLKHO  $\bowtie$  MaCH (CUAHANG2)

KHOHANG\_NV1 = KHOHANG\_NV  $\bowtie$  MaCH (CUAHANG1)

KHOHANG\_NV2 = KHOHANG\_NV  $\bowtie$  MaCH (CUAHANG2)

HOADON1 = HOADON  $\bowtie$  MaNV (NHANVIEN1)

HOADON2 = HOADON  $\bowtie$  MaNV (NHANVIEN2)

CTHD1 = CTHD  $\bowtie$  MaHD (HOADON1)

CTHD2 = CTHD  $\bowtie$  MaHD (HOADON2)

- Quan hệ SUATUOI, KHACHHANG được nhân bản ở cả 2 cửa hàng.

### B. Câu truy vấn đơn giản

```
SELECT NV.MaNV, NV.MaCH, NV.TenNV,
SUM(QLK.SoLuong)
FROM NHANVIEN NV
JOIN KHOHANG_QLKHO QLK ON QLK.MaCH =
NV.MaCH
JOIN KHOHANG_NV KHN ON KHN.MaCH =
NV.MaCH
WHERE QLK.NgayNhap > '02-DEC-23' AND
NV.GioiTinh = 'Nữ' AND KHN.TinhTrang = 'Hết
hàng'
GROUP BY NV.MaNV, NV.MaCH, NV.TenNV
```

### C. EXPLAIN Query câu truy vấn đơn giản chưa tối ưu

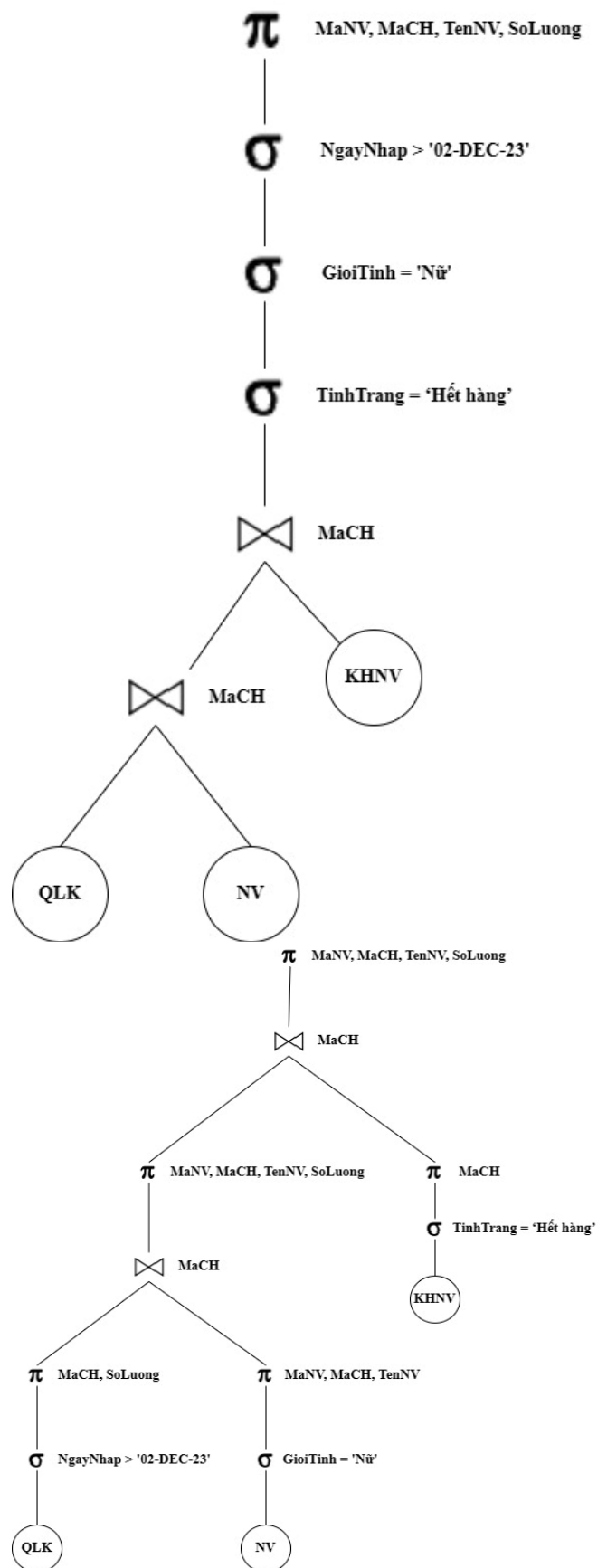
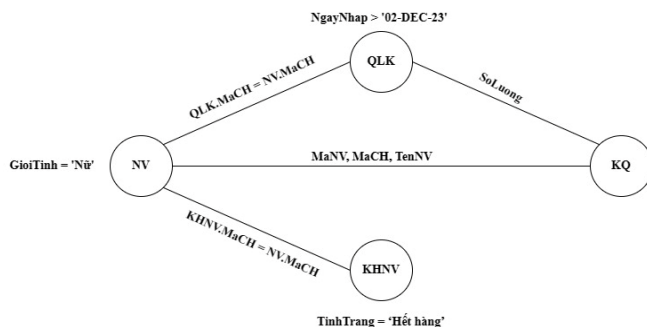
```
EXPLAIN PLAN FOR SELECT NV.MaNV,
NV.MaCH, NV.TenNV, SUM(QLK.SoLuong)
FROM NHANVIEN NV
JOIN KHOHANG_QLKHO QLK ON QLK.MaCH =
NV.MaCH
JOIN KHOHANG_NV KHN ON KHN.MaCH =
NV.MaCH
WHERE QLK.NgayNhap > '02-DEC-23' AND
NV.GioiTinh = 'Nữ' AND KHN.TinhTrang = 'Hết
hàng'
GROUP BY NV.MaNV, NV.MaCH, NV.TenNV
```

PLAN\_TABLE\_OUTPUT

1 Plan hash value: 2220553367

Id	Operation	Name	Rows	Bytes	Cost (CPU)	Time
0	SELECT STATEMENT		13	975	9 (12)	00:00:01
1	HASH GROUP BY		13	975	9 (12)	00:00:01
2	HASH JOIN		16	1200	8 (0)	00:00:01
3	HASH JOIN		5	235	5 (0)	00:00:01
4	VIEW	VM_GSC_10	1	19	2 (0)	00:00:01
5	HASH GROUP BY		1	18	2 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID BATCHED	KHOHANG_QLKHO	4	72	2 (0)	00:00:01
7	INDEX FULL SCAN	PK_KHOHANG_QLKHO	10	1	1 (0)	00:00:01
8	TABLE ACCESS FULL	NHANVIEN	5	190	3 (0)	00:00:01
9	TABLE ACCESS FULL	KHOHANG_NV	4	72	3 (0)	00:00:01

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



- Câu truy vấn sau khi tối ưu hóa

```
SELECT MaNV, MaCH, TenNV, SUM(SoLuong)
FROM (
SELECT A1.MaNV, A1.MaCH, A1.TenNV, (SoLuong)
FROM (SELECT MaNV, MaCH, TenNV
```



```

FROM NHANVIEN WHERE GioiTinh = 'Nữ') A1
JOIN (SELECT MaCH, SoLuong
FROM KHOHANG_QLKHO WHERE NgayNhap >
'02-DEC-23') A2 ON A1.MaCH = A2.MaCH
JOIN (SELECT TinhTrang, MaCH
FROM KHOHANG_NV WHERE TinhTrang = 'Hết
hàng') A3 ON A2.MaCH = A3.MaCH
)
GROUP BY MaNV, MaCH, TenNV

```

- Explain query:

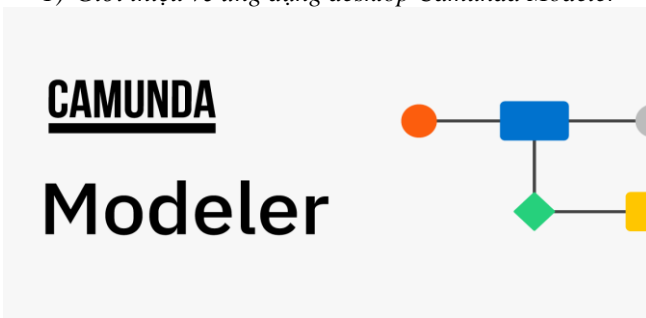
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		13	975	8 (13)	00:00:01
1	HASH GROUP BY		13	975	8 (13)	00:00:01
2	HASH JOIN		16	1200	7 (0)	00:00:01
3	NESTED LOOPS		4	148	4 (0)	00:00:01
4	NESTED LOOPS		10	148	4 (0)	00:00:01
5	VIEW	VW_GBC_10	1	19	2 (0)	00:00:01
6	HASH GROUP BY		1	18	2 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID BATCHED	KHOHANG_QLKHO	4	72	2 (0)	00:00:01
8	INDEX FULL SCAN	PK_KHOHANG_QLKHO	10		1 (0)	00:00:01
9	INDEX RANGE SCAN	PK_KHOHANG_NV	1		2 (0)	00:00:01
10	TABLE ACCESS BY INDEX ROWID	KHOHANG_NV	4	72	2 (0)	00:00:01
11	TABLE ACCESS FULL	NHANVIEN	5	190	3 (0)	00:00:01

## VII. KẾT NỐI ỨNG DỤNG APPLICATION WINDOWS HOẶC WEB THỰC TẾ

### A. Giới thiệu chung

Dựa theo kiến thức đã học ở môn EC201 - Phân tích thiết kế quy trình nghiệp vụ doanh nghiệp, em kết hợp thực hiện việc kết nối ứng dụng Camunda Modeler với hệ quản trị cơ sở dữ liệu Oracle để thực hiện lệnh truy vấn tới các bảng dữ liệu

#### 1) Giới thiệu về ứng dụng desktop Camunda Modeler



Đây là một công cụ mạnh mẽ được thiết kế để hỗ trợ trong quá trình quản lý và tối ưu hóa quy trình kinh doanh. Được xây dựng trên nền tảng Camunda BPMN (Business Process Model and Notation), công cụ này mang đến cho người dùng khả năng thiết kế, triển khai, và theo dõi các quy trình kinh doanh một cách hiệu quả.

### B. Cấu hình

#### 1) Cấu hình Bot

```

import { Client, logger, Variables } from
"camunda-external-task-client-js";
import connection from
"./data/connecto.js";
import dotenv from 'dotenv';
dotenv.config()
const config = {
  baseUrl:
"http://localhost:8080/engine-rest",
  use: logger,
};
// Khởi tạo đối tượng Client
const client = new Client(config);

```

```

// Định nghĩa hàm xử lý cho external task
với key là "oraclebot"
const registerClaimHandler = async ({
  task, taskService }) => {
  // Lấy dữ liệu từ biến của
  external task
  const CUSID =
task.variables.get("CUSTOMERID");
  const CUSNAME =
task.variables.get("NAME");
  const PHONE =
task.variables.get("PHONE");
  const ADDRESS =
task.variables.get("ADDRESS");
  console.log(task.variables);
  try {
    // Thực hiện INSERT dữ liệu vào
    bảng "KHACHHANG" trong cơ sở dữ liệu
    Oracle
    const data = await
connection.execute(`INSERT INTO KHACHHANG
VALUES ('${CUSID}', '${CUSNAME}',
'${PHONE}', '${ADDRESS}')`);
    console.log(data.rows);
    // Thực hiện COMMIT để xác nhận
    các thay đổi và Lưu trữ dữ liệu
    await
connection.execute("COMMIT");
    console.log("Dữ liệu đã được cập
nhật thành công");
    await taskService.complete(task);
    console.log("Added new claim to
database");
  } catch (e) {
    // Nếu có lỗi, thực hiện ROLLBACK
    để hủy bỏ các thay đổi
    await connection.rollback();
    throw new Error(`Failed completing
register claim task: ${e}`);
  } finally {
    // Cuối cùng, đóng kết nối
    await connection.close();
  }
};
client.subscribe("oraclebot",
registerClaimHandler);

```

#### 2) Cấu hình Connection

```

import oracledb from 'oracledb';
import dotenv from 'dotenv';
dotenv.config();
// Thiết lập thông tin kết nối Oracle
const oracleConfig = {
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  connectString:
`${process.env.DB_HOST}:${process.env.PORT}
/${process.env.DB_NAME}`,
};
// Kết nối và trả về promise
async function connectToOracle() {

```

```

let connection;
try {
  connection = await
oracledb.getConnection(oracleConfig);
  console.log('Kết nối Oracle thành
công');
  return connection;
} catch (error) {
  console.error('Lỗi kết nối Oracle:',
error);
  throw error;
}
}
// Kết nối và trả về promise
const connection = await
connectToOracle();
export default connection;

```

### 3) Cấu hình file dotenv

```

# Connect oracle
DB_HOST = '26.182.238.200'
DB_USER = 'CH2'
DB_PASSWORD = 'CH2'
DB_NAME = 'orcl'
PORT = 1521

```

## VIII. LỜI CẢM ƠN

Trước tiên, chúng tôi muốn dành những lời tri ân đặc biệt đến Thầy Nguyễn Minh Nhựt, một người giảng viên đã dày công truyền đạt tri thức và lòng đam mê của mình cho chúng tôi trong khóa học Cơ sở dữ liệu phân tán.

Những kiến thức mà Thầy truyền đạt đã trở thành tài sản vô giá, giúp chúng tôi vượt qua khó khăn và hoàn thành đề tài đồ án một cách xuất sắc hơn. Dựa trên những kiến thức được Thầy truyền đạt và sự cống hiến của cả nhóm trong việc tự nghiên cứu và khám phá các công cụ và kiến thức mới, chúng tôi đã dành thời gian để thiết kế một cơ sở dữ liệu phân tán.

Trong quá trình thực hiện đồ án, chúng tôi đã nỗ lực hết mình để đạt được kết quả tốt nhất. Tuy nhiên, không tránh khỏi những sai sót và hạn chế do sự hạn hẹp về thời gian và khả năng cá nhân của từng thành viên trong nhóm. Đồ án của chúng tôi, mặc dù đã có những nỗ lực, vẫn chỉ là một sản phẩm mang tính chất đồ án học tập và chưa thực sự hoàn thiện.

Chúng tôi trân trọng mong nhận được những góp ý giá trị từ Thầy, nhằm hoàn thiện những kiến thức đã được học và đồng thời làm bổ sung cho chúng tôi trong các dự án và công việc sắp tới. Chúng tôi biết ơn và trân trọng sự chia sẻ kiến thức và kỹ năng từ Thầy, đã giúp mỗi thành viên trong nhóm nâng cao vốn kiến thức hiện có, để chúng tôi có thể tiếp tục hoàn thiện các dự án và sản phẩm khác trong tương lai.

Một lần nữa, chúng tôi muốn gửi lời cảm ơn chân thành và tốt đẹp nhất đến Thầy cô và các bạn trong lớp!

## TÀI LIỆU THAM KHẢO

- [1] "Database link with Radmin VPN", tài liệu thực hành Cơ sở dữ liệu phân tán, Trường Đại học Công nghệ thông tin
- [2] ThS. Trình Trọng Tín tài liệu thực hành môn Phân tích thiết kế quy trình nghiệp vụ doanh nghiệp
- [3] ThS. Thái Bảo Trân, "Chương 5: Tối ưu hoá truy vấn trong CSDL phân tán", tài liệu lý thuyết môn Cơ sở dữ liệu phân tán, Trường Đại học Công nghệ thông tin
- [4] ThS. Thái Bảo Trân, "Chương 6: Quản lý giao tác và điều khiển đồng thời phân tán", tài liệu lý thuyết môn Cơ sở dữ liệu phân tán, Trường Đại học Công nghệ thông tin.
- [5] Ths. Thái Bảo Trân, "Đề thi cuối kỳ môn CSDLPT học kỳ 2 2019-2020", đề thi nội bộ Trường Đại học Công nghệ thông tin.
- [6] Karen Morton, Kerry Osborne, Robyn Sands, Riyaj Shamsudeen and Jared Still, "Pro Oracle SQL – Second Edition" - 2013