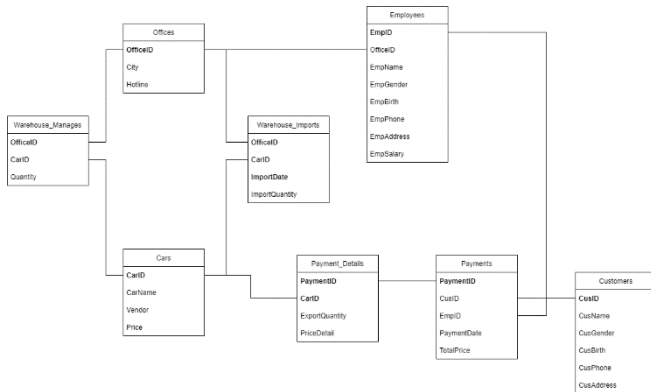


# Thiết Kế Cơ Sở Dữ Liệu Phân Tán

## I. TỔNG QUAN

- 1) Giới thiệu chung
- 2) Động lực nghiên cứu
- 3) Phạm vi báo cáo
- 4) Thách thức
- 5) Đóng góp của báo cáo
- 6) Cấu trúc báo cáo

## II. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN TRÊN MÔI TRƯỜNG MÁY ẢO, RADMIN



- 1) Thiết kế cơ sở dữ liệu
- 2) Thực hiện 10 câu truy vấn
  - 2.1) Liệt kê nhân viên (*EmpID*, *EmpName*, *EmpSalary*) ở cả 2 chi nhánh (Hội)
  - 2.2) Liệt kê 5 hóa đơn có giá trị cao nhất, sắp xếp giảm dần theo giá trị (*PaymentID*, *Tong*, *SL*) (Gom nhóm)
  - 2.3) Cho biết chi nhánh nào bán được đa dạng xe nhất (*OfficeID*, *City*, *SLLoại*) (Tính toán)
  - 2.4) Tìm khách hàng đã mua xe ở cả hai chi nhánh (*CusID*, *CusName*) (Giao)
  - 2.5) Tìm khách hàng chỉ mua xe ở chi nhánh *Office1* (*CusID*, *CusName*) (Trừ)
  - 2.6) Tìm loại xe được mua nhiều nhất (*CarID*, *CarName*, *SL*) (Gom nhóm)
  - 2.7) Xem số lượng xe hiện có ở cả hai chi nhánh (*CarName*, *SLOffice1*, *SLOffice2*)
  - 2.8) Cho biết tổng số lượng xe đã nhập, đã xuất của từng loại xe trong năm 2020 của chi nhánh *Office2* (*CarName*, *SLNhap*, *SLXuat*)
  - 2.9) Tìm nhân viên bán được nhiều xe nhất trong năm 2016 của cả hai chi nhánh
  - 2.10) Tìm hóa đơn mua tất cả loại xe hiện có trong công ty có nhà cung cấp là Toyota. (Phép chia)

## III. VIẾT HÀM, THỦ TỤC, RÀNG BUỘC TOÀN VỆN TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

- 1) Trigger
  - Trigger kiểm tra thông tin xe nhập vào phải có giá trị lớn hơn 0.

- Trigger kiểm tra nhân viên phải đạt tối thiểu 18 tuổi khi vào làm việc
- 2) Procedure
    - Tìm nhân viên được nhập từ máy và thay đổi mức lương của nhân viên đó theo mức lương được nhập vào
  - 3) Function
    - Tính tổng tiền tất cả các hóa đơn khách hàng chi trả

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

- 1) Lost Update
- 2) Non-Repeatable
- 3) Deadlock
- 4) Dirty Read
- 5) Phantom Read

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

### 1) Lược đồ phân mảnh

- Quan hệ OFFICES phân mảnh ngành chính theo “City”:

OFFICE1= σCity= ‘New York’ (OFFICES)

OFFICE2= σCity= ‘Paris’ (OFFICES)

- Quan hệ EMPLOYEES, PAYMENTS, PAYMENT\_DETAILS, WAREHOUSE\_MANAGES, WAREHOUSE\_IMPORTS phân mảnh ngang dẫn xuất như sau:

EMPLOYEES1=EMPLOYEES⋈OfficeID (OFFICES1)

EMPLOYEES2= EMPLOYEES⋈OfficeID (OFFICES2)

PAYMENTS1=PAYMENT ⋈EmpID (EMPLOYEES1)

PAYMENTS2= PAYMENT ⋈EmpID (EMPLOYEES2)

PAYMENT\_DETAILS1=PAYMENT\_DETAILS ⋈EmpID (PAYMENT1)

PAYMENT\_DETAILS2= PAYMENT\_DETAILS ⋈EmpID (PAYMENT2)

WAREHOUSE\_MANAGES1=WAREHOUSE\_MANAGES⋈OfficeID (OFFICES1)

WAREHOUSE\_MANAGES2=WAREHOUSE\_MANAGES⋈OfficeID (OFFICES2)

WAREHOUSE\_IMPORTS1=WAREHOUSE\_IMPORTS⋈OfficeID (OFFICES1)

WAREHOUSE\_IMPORTS2=WAREHOUSE\_IMPORTS⋈OfficeID (OFFICES2)

Quan hệ CARS, CUSTOMERS được nhân bản ở cả hai chi nhánh.

- Câu truy vấn cần tối ưu:

Thống kê số lượng xe đã bán được của từng nhân viên ở Office1, giảm dần theo số lượng.

- Thực hiện Explain query

EXPLAIN PLAN FOR

SELECT E.EMPID, EMPNAME,  
SUM(ExportQuantity) AS DOANHSO

FROM OFFICES O, EMPLOYEES E, PAYMENTS P,  
PAYMENT\_DETAILS PD

WHERE O.OFFICEID=E.OFFICEID

AND E.EMPID=P.EMPID

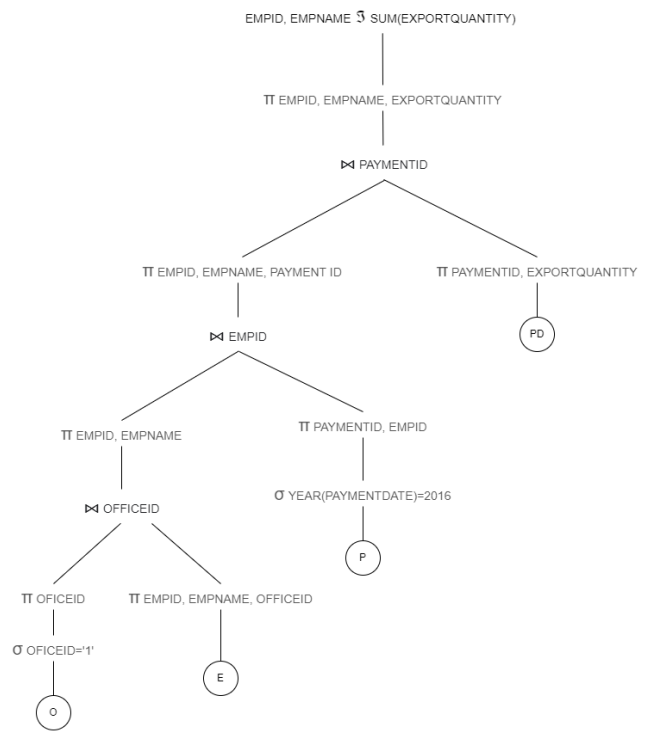
AND P.PAYMENTID=PD.PAYMENTID

AND EXTRACT(YEAR FROM  
PAYMENTDATE)=2016

AND O.OFFICEID='1'

GROUP BY E.EMPID, EMPNAME

ORDER BY DOANHSO DESC;

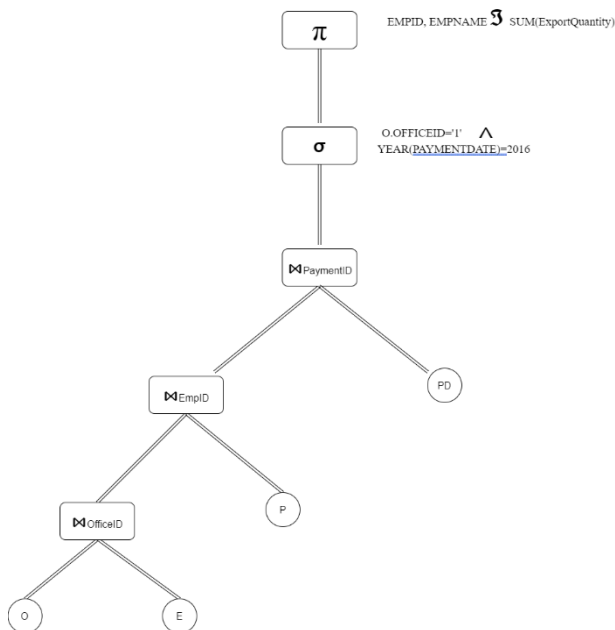


- Cây truy vấn dựa vào lược đồ phân mảnh

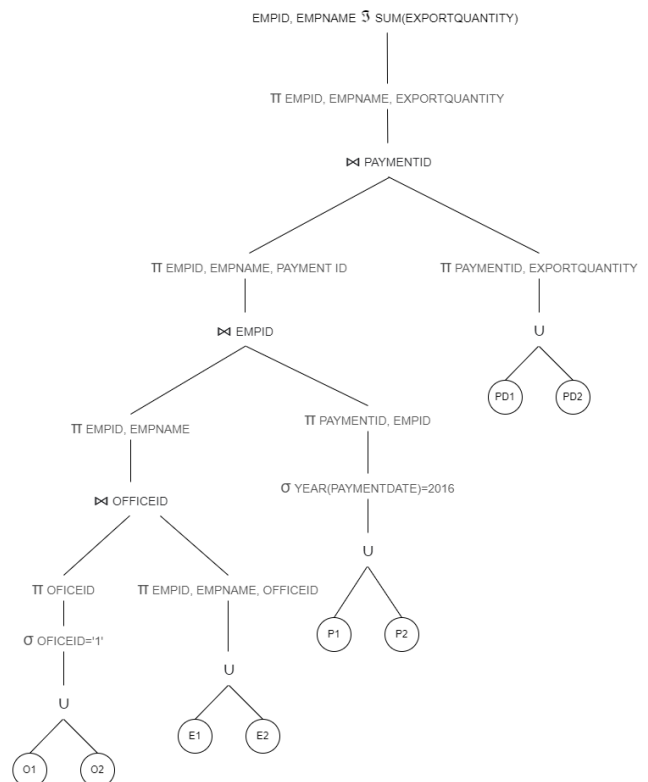
select \* from table(dbms\_xplan.display);

2) Tối ưu hóa cây truy vấn

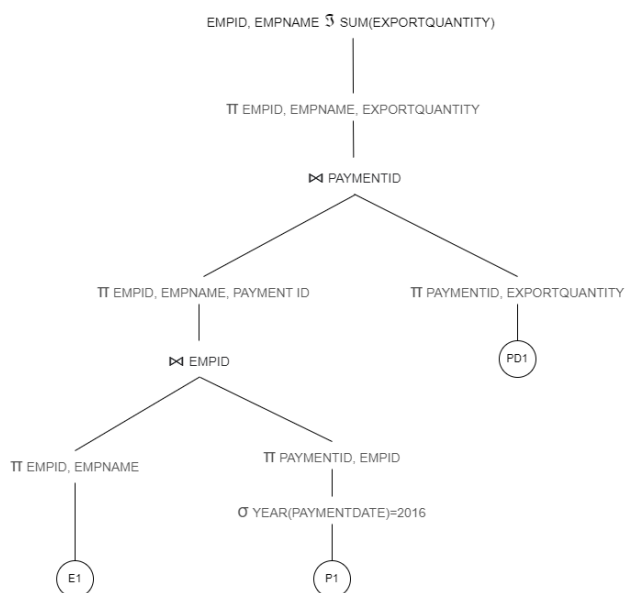
- Cây truy vấn quan hệ ban đầu



- Cây truy vấn sau khi tối ưu hóa toàn cục



- Cây truy vấn sau khi tối ưu hóa



- Câu truy vấn sau khi tối ưu hóa

```
SELECT EMPID, EMPNAME,
SUM(EXPORTQUANTITY) AS SL
```

```
FROM(
```

```
SELECT EMPID, EMPNAME, EXPORTQUANTITY
```

```
FROM
```

```
(SELECT E1.EMPID, EMPNAME, PAYMENTID
```

```
FROM
```

```
(SELECT EMPID, EMPNAME
```

```
FROM EMPLOYEES) E1
```

```
JOIN
```

```
(SELECT PAYMENTID, EMPID
FROM PAYMENTS
WHERE EXTRACT(YEAR FROM
PAYMENTDATE)=2016) P1
```

```
ON E1.EMPID=P1.EMPID
```

```
) EP1
```

```
JOIN
```

```
(SELECT PAYMENTID, EXPORTQUANTITY
```

```
FROM PAYMENT_DETAILS
```

```
) PD1
```

```
ON EP1.PAYMENTID=PD1.PAYMENTID)
```

```
GROUP BY EMPID, EMPNAME;
```

## VI. CƠ CHẾ NHẬN BẢN TRONG ORACLE

- 1) Nhân bản Oracle là gì?
- 2) Nhân bản Oracle để làm gì?
- 3) Nhân bản Oracle hoạt động như thế nào?
- 4) Các phương pháp nhân bản cơ sở dữ liệu Oracle
  - 4.1) Kết xuất đầy đủ và tải (Full Dump and Load)
  - 4.2) Incremental Approach (Table Differencing)
  - 4.3) Trigger-Based Approach
  - 4.4) Change Data Capture
- 5) Demo