CHUYÊN ĐỀ ORACLE

Giáo viên : Trần Thị Ngọc Châu

Nội dung chính

1. Giới thiệu về Oracle
2. Sử dụng PL/SQL
3. Sử dụng công cụ From
4. Sử dụng công cụ Report

TRẦN THỊ NGỌC CHÂU

1.Giới thiệu về ORACLE

- Oracle?
- So sánh giữa Oracle và SQL server
- Khác nhau giữa SQL và PL/SQL

1.1 ORACLE ?

- Là hệ quản trị cơ sở dữ liệu
- Tập hợp các sản phẩm phục vụ cho mục đích xây dựng các ứng dụng
- Cung cấp các công nghệ thông tin hoàn chỉnh

1.1 ORACLE ?

Tại sao phải dùng Oracle?

- Oracle là một hệ thống mở
- Xây dựng cơ sở dữ liệu ở mọi kích cở
- Hỗ trợ cho nhiều người sử dụng đồng thời
- Hạn chế sự tranh chấp dữ liệu
- Đảm bảo sự thống nhất dữ liệu
- Hỗ trợ môi trường Client/server thật sự
- Tương thích với nhiều flatform (Windows, Unix…

TRẦN THỊ NGỌC CHÂU

1.1 ORACLE ?

Sản phẩm Oracle

- **Oracle card** : Tạo các ứng dụng dựa trên CSDL có kết hợp các đặc tính đồ họa
- **Oracle case** : Trợ giúp phân tích, thiết kế, tạo ứng dụng
- **SQL*Plus**: dùng ngôn ngữ SQL để tương tác với CSDL trên oracle
- **Pro *Oracle** : Tập hợp các tính biên dịch cho phép truy xuất csdl oracle bằng các ngôn ngữ lập trình
- **Oracle Text retrieval**: Cung cấp thêm các chức năng tìm kiếm văn bản trên Oracle

TRẦN THỊ NGỌC CHÂU

SO SÁNH GIỮA ORACLE VÀ SQL SERVER

- Oracle lớn hơn SQL Sever nhiều lắm, một tablespace trong Oracle tương đương 1 datbase trong SQL Sever
- Oracle là một RDBMS multiplatform trong khi SQL Server chỉ giới hạn trên NT Server. Hầu hết các hệ Oracle high-end chạy trên UNIX.
- Oracle không có khái niệm một CSDL master. Tất cả CSDL chạy độc lập với các file dữ liệu của nó, sự quản lý bộ nhớ riêng và điều khiển riêng.
- Kiến trúc của Oracle hoàn toàn khác SQL Server (Cái này để trantuananh24hg giải thích rõ hơn)

SO SÁNH GIỮA ORACLE VÀ SQL SERVER

- Oracle không được tích hợp vào Windows như SQL Server
- SQL Server sử dụng mở rộng Transact-SQL cho SQL, Oracle thì sử dụng PL/SQL.
- Các **stored procedure** SQL Server trả về một **Recordset** nếu bạn làm một câu lệnh SELECT trong procedure. Oracle chỉ hỗ trợ điều này qua cursor variables.
- Trong các stored procedure, Oracle tự động sử dụng các chuyển tác, trong SQL Server sự thay đổi dữ liệu được tự động commit theo mặc định SQL Server chia sẽ khái niệm cột autonumber với Access. Trong Oracle, bạn sẽ làm việc với các sequence.

SO SÁNH GIỮA ORACLE VÀ SQL SERVER

- SQL Server hỗ trợ các bảng tạm, Oracle không có.
- Trong Oracle không phải debug giữa client và server như trong SQL Server.
- Các hàm khác nhau giữa hai hệ thống, và một số hàm không có hàm tương ứng SQL Server có một tập kiểu dữ liệu cơ sở lớn hơn Oracle
- Oracle không hỗ trợ cursor server-side.
- Oracle sử dụng lock mức hàng, trong khi trước version 7, SQL Server sử dụng lock mức trang.
- Giống nhau:
  - Cả hai sản phẩm hỗ trợ SQL và các stored procedure.
  - Chức năng của những ngôn ngữ này tương tự nhau,nhưng khác nhau về cú pháp

KHÁC NHAU GIỮA SQL VÀ PL/SQL

SQL	SQL*Plus
• Là ngôn ngữ để giao tiếp với Oracle Server trong việc truy xuất dữ liệu.	• Nhận dạng lệnh SQL và gửi lệnh lên Server .
•Câu lệnh dựa trên bộ ký tự chuẩn ASCII	•Tuỳ thuộc vào từng phiên bản của Oracle Không
•Thao tác trên các dữ liệu có trong các bảng đã được định nghĩa trong database	• thao tác với dữ liệu trong database
•Câu lệnh được nạp vào bộ nhớ đệm trên một hoặc nhiều dòng	•Câu lệnh được tải trực tiếp không thông qua bộ đệm
•Câu lệnh không được viết tắt	•Câu lệnh có thể viết tắt
•Có sử dụng ký tự kết thúc lệnh khi thực hiện	• Không đòi hỏi phải có ký tự kết thúc lệnh
•Sử dụng các hàm trong việc định dạng dữ liệu	•Sử dụng các lệnh định dạng dữ liệu của chính

CÚ PHÁP LỆNH

- **SELECT [DISTINCT] {*, column [alias],...}**
- **FROM table;**

Câu lệnh truy vấn cơ bản : cả SQL và PL/SQL giống nhau

Ví dụ : SELECT * FROM CUSTOMERS;

CÚ PHÁP LỆNH

Trong cú pháp cơ bản nêu trên còn có thể đưa vào các thành phần khác nhau:

- Biểu thức toán học
- Column alias
- Các column được ghép chuỗi Literal

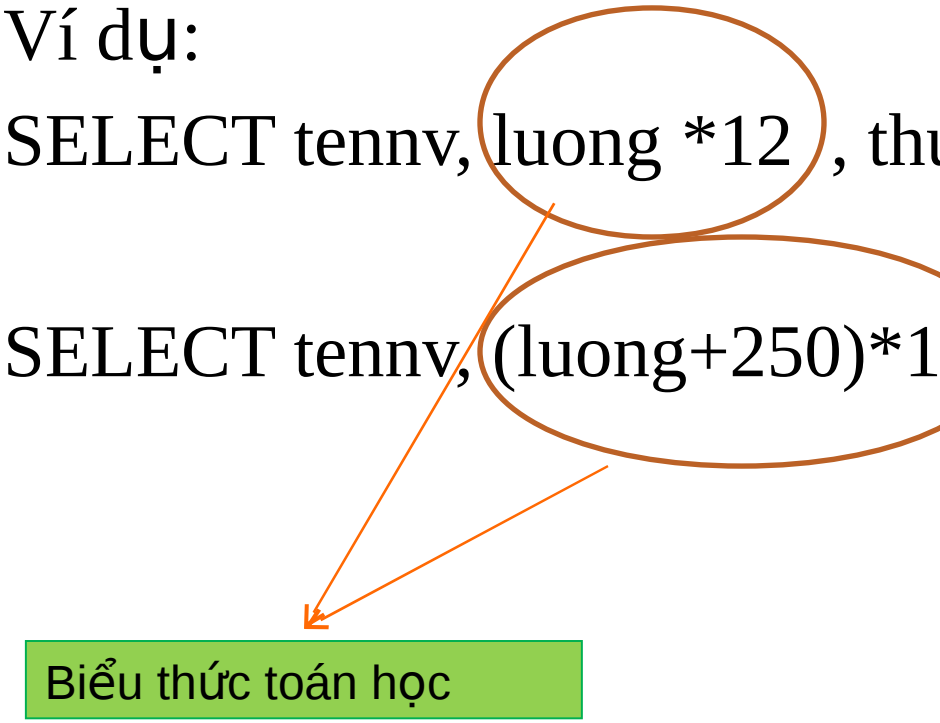
CÚ PHÁP LỆNH

- **Biểu thức toán học : (+, -, *, /)**

Ví dụ:

SELECT tennv, luong *12 , thuong FROM nhanvien;

SELECT tennv, (luong+250)*12 FROM nhanvien;



KHÁC NHAU GIỮA SQL VÀ PL/SQL

CÚ PHÁP LỆNH

- **Column alias** : là phần nhãn hiện thị cho tiêu đề Column

Ví dụ: (luongnam chính là column alias)

Câu lệnh PL/SQL

SELECT tennv, **luong*12 "lương năm"**, thuong

Cả 2 đều khác nhau

Câu lệnh của SQL :

```
SELECT tennv,luong*12 [lương năm ], thuong
FROM nhanvien
```

KHÁC NHAU GIỮA SQL VÀ PL/SQL

CÚ PHÁP LỆNH

- Ghép tiếp các cột dữ liệu :Toán tử ghép tiếp chuỗi (||)

Ví dụ: **PL/SQL**

```
SELECT manv || tennv “nhân viên”
FROM nhanvien;
```

Cả 2 đều khác nhau

SQL

```
SELECT manv + tennv “nhân viên”
FROM nhanvien;
```

CÚ PHÁP LỆNH

- Ghép tiếp chuỗi ký tự

Ví dụ: **PL/SQL**

```
SELECT manv || tennv || 'làm việc trong phòng '
        || mapb "chi tiet nhan vien"
FROM nhanvien;
```

SQL

```
SELECT manv + tennv + 'là việc trong phòng'
+ mapb [Employee Detail]
FROM nhanvien
```

KẾT LUẬN

- Lưu trữ dữ liệu thì Oracle lưu một khối lượng rất lớn
- Trong SQL cách đặt nhãn cho Column là dùng từ khóa as và có khoảng cách tên nhãn là dấu [ ].Con oracle la dấu nháy kép “ ”.
- Nối hai chuỗi thì SQL là dấu “+” , còn Oracle là dấu “||”
- Câu lệnh trong SQL không viết tắt được nhưng trong Oracle thì được.

2.Sử DụNG PL/SQL

- Truy vấn PL/SQL
- Tạo CSDL
- Cập nhật CSDL
- Lập trình PL/SQL

TRUY VẤN PL/SQL

TRẦN THỊ NGỌC CHÂU

NỘI DUNG CHÍNH

- LỆNH TRUY VẤN CƠ BẢN
- TRUY VẤN CÓ ĐIỀU KIỆN
- THỰC THI CÂU LỆNH VỚI CÁC BIẾN THAY THẾ
- CÁC LOẠI HÀM
- THÁO TÁC TRÊN MỘT NHÓM
- TRUY VẤN CON

TRUY VẤN CƠ BẢN

Loại bỏ các dòng trùng nhau

Thành phần cơ bản

SELECT [DISTINCT] {*,COLUN [ALIAS]}

Bí danh cho cột

FROM TABLE

|| : kết nối các cột

ORDER BY {COLUMN,EXPR} [ASC| DESC]

TRẦN THỊ NGỌC CHÂU

TRUY VẤN CƠ BẢN

- Ví dụ: User scott
- **SELECT** ENAME,JOB,SAL*12 LUONGNAM,DEPTNO
- FROM EMP
- **ORDER BY** ENAME

	ENAME	JOB	LUONGNAM	DEPTNO
1	ADAMS	CLERK	13200	20
2	ALLEN	SALESMAN	19200	30
3	BLAKE	MANAGER	34200	30
4	CLARK	MANAGER	29400	10
5	FORD	ANALYST	36000	20

TRUY VẤN CƠ BẢN

- **Toán tử logic : >, >=, <, <=, =**
- **Toán tử SQL:**

TOÁN TỬ	Ý NGHĨA
BETWEEN… AND…	Giữa 2 giá trị
IN	Tương ứng 1 vài giá trị trong danh sách
LIKE	Tìm kiếm gần đúng
IS NULL	Giá trị rỗng

TRUY VẤN CÓ ĐIỀU KIỆN

TRUY VẤN CÓ ĐIỀU KIỆN

Cú pháp

- SELECT [DISTINCT] {*, column [alias],...}
  FROM table
- [**WHERE condition (s)**];

Truy vấn có điều kiện

- **Toán tử SQL:**
- **Ví dụ : tìm tên nhân viên bắt đầu bởi ký tự S**

```
SELECT ENAME,EMPNO
FROM EMP
WHERE ENAME LIKE 'S%'
```

TRUY VẤN CÓ ĐIỀU KIỆN

- Ví dụ: liệt kê danh sách lương trong khoảng từ 1000 đến 2000

```
SELECT DEPTNO, JOB, ENAME,
  SAL
FROM EMP
WHERE SAL BETWEEN 1000 AND
  2000 ;
```

TRUY VẤN CÓ ĐIỀU KIỆN

Các toán tử sử dụng trong mệnh đề where

- **[NOT] BETWEEN x AND y** : [Không] lớn hơn hoặc bằng x và nhỏ hơn hoặc bằng y
- **IN (danh sách):** Thuộc bất kỳ giá trị nào trong danh sách
- **x [NOT] LIKE y** : Đúng nếu x [không] giống khung mẫu y
- **IS [NOT] NULL** : Kiểm tra giá trị rỗng

TRUY VẤN CÓ ĐIỀU KIỆN

Các toán tử sử dụng trong mệnh đề where

- **EXISTS** : Trả về TRUE nếu có tồn tại
- **NOT**: Phủ định mệnh đề
- **AND** : Yêu cầu dữ liệu phải thoả mãn cả 2 điều kiện
- **OR** : Cho phép dữ liệu thoả mãn 1 trong 2 điều kiện

TRUY VẤN CÓ ĐIỀU KIỆN

- Ví dụ 1:

IN (danh sách)

Chọn nhân viên có lương bằng một trong 2 giá trị 1400 hoặc 3000

- **SELECT** * **FROM** emp **WHERE** sal **IN** (1400, 3000);

TRUY VẤN CÓ ĐIỀU KIỆN

x [NOT] LIKE y

- Tìm nhân viên có tên bắt đầu bằng chuỗi SMITH

```
SELECT * FROM emp
WHERE  ename LIKE 'SMITH_';
```

- Để chọn những nhân viên có tên bắt đầu bằng 'SM'

```
SELECT * FROM emp
WHERE ename LIKE 'SM%';
```

NỘI DUNG CHÍNH

THỰC THI CÂU LỆNH VỚI CÁC BIẾN THAY THẾ

THỰC THI CÂU LỆNH VỚI CÁC BIẾN THAY THẾ

- Biến thay thế với dấu **&**
  - **Biến kiểu số :**

Ví dụ :

```
select empno, ename,  sal
from emp
where deptno = &deptnumber ;
```

→**Thực thi : hiển thị yêu cầu**

Nhập giá trị deptnumber: 10

THỰC THI CÂU LỆNH VỚI CÁC BIẾN THAY THẾ

- Biến thay thế với dấu **&**
  - **Biến kiểu chuỗi :**

Ví dụ :

```
select empno, ename,  sal *12 luongnam
from emp
where job = '&job_title' ;
```

- **Thực thi : hiển thị yêu cầu**

Nhập giá trị job_title= MANAGER

THỰC THI CÂU LỆNH VỚI CÁC BIẾN THAY THẾ

- Biến thay thế với dấu **&**
  - **Đặt biến cho tên cột bằng một biểu thức**

Ví dụ :

```
select empno, ename,job,  &bieuthuc
from emp ;
```

- **Thực thi : hiển thị yêu cầu**

Nhập giá trị bieuthuc= sal/2

NỘI DUNG CHÍNH

CÁC LOẠI HÀM

TRẦN THỊ NGỌC CHÂU

CÁC LOẠI HÀM

- **Hàm dùng cho kiểu số**

HÀM	CÚ PHÁP	GIẢI THÍCH
ABS	ABS(col/value)	Giá trị tuyệt đối
FLOOR	FLOOR(col/value)	Trả về số nguyên gần nó nhất
MOD	MOD(m,n)	Trả về số dư
ROUND	ROUND(col/value,n)	Hàm làm tròn
SIGN	SIGN(col/value)	Trả về số âm
TRUNC	TRUNCT(col/value,n)	Cắt ngắn tới n số lẻ

CÁC LOẠI HÀM

- **Hàm dùng cho kiểu chuỗi**

HÀM	CÚ PHÁP	GIẢI THÍCH
CONCAT	CONCAT(char1,char2)	Nối chuỗi 2 vào chuỗi 1
INITCAP	INITCAP(col/value)	Trả về ký tự đầu hoa
LOWER/UPPER	LOWER(Col/value)	Trả về chuỗi thường/hoa
LTRIM/ RTRIM	LTRIM(col/value)	Bỏ bớt các khoảng trắng bên trái/phải

CÁC LOẠI HÀM

Hàm dùng cho kiểu chuỗi

HÀM	CÚ PHÁP	GIẢI THÍCH
CONCAT	CONCAT(char1,char2)	Nối chuỗi 2 vào chuỗi 1
INITCAP	INITCAP(col/value)	Trả về ký tự đầu hoa
LOWER/UPPER	LOWER(Col/value)	Trả về chuỗi thường/hoa
ROUND	ROUND(col/value,n)	Hàm làm tròn
SIGN	SIGN(col/value)	Trả về số âm
LTRIM/ RTRIM	LTRIM(col/value)	Bỏ bớt các khoảng trắng bên trái/phải

CÁC LOẠI HÀM

Hàm dùng cho kiểu ngay

HÀM	MỤC ĐÍCH
SYSDATE	Lấy ngày hệ thống
MONTH_BETWEEN (date1,date2)	Trả về số tháng giữa d1 và d2
ADD_MONTHS(date,n)	Cộng n tháng tới date
NEXT_DAY(date,char/number)	Chỉ ra ngày sau char/number sau date
LAST_DATE(date)	Chỉ ra ngày sau cùng trong tháng của date

CÁC LOẠI HÀM

Hàm dùng chuyển kiểu

HÀM	MỤC ĐÍCH
TO_CHAR(num/char, 'frm')	Được thay đổi đến dạng chỉ định
TO_NUMBER(char)	Chuyển kiểu chuỗi sang số
TO_DATE(char,'frm')	Trình bày ngày theo dạng format chỉ định

CÁC LOẠI HÀM

- **Hàm dùng cho mọi kiểu dữ liệu**

1. **DECODE(col/exp,search1,result1, [s1,r1….] default)**

Kết quả : trả về result nếu col/epx bằng với search1, ngược lại thì theo giá trị default

CÁC LOẠI HÀM

- **Hàm dùng cho mọi kiểu dữ liệu**

2. NVL(cal/exp,val)

Kết quả : : biến đổi giá trị null thành giá trị

Ví dụ :

```
select sal*12+nvl(comm,0),nvl(comm,1000)
from emp where deptno=10;
```

CÁC LOẠI HÀM

- **Hàm dùng cho mọi kiểu dữ liệu**

2. GREATEST(Col/value,col/value…)

Kết quả : : giá trị lớn nhất trong danh sách

```
SELECT GREATEST(1000,2000),GREATEST(SAL, COMM)
FROM   EMP;
```

CÁC LOẠI HÀM

- **Hàm dùng cho mọi kiểu dữ liệu**

2. LEAST(Col/value,col/value…)

Kết quả : : giá trị nhỏ nhất trong danh sách

```
SELECT GREATEST(1000,2000),GREATEST(SAL, COMM)
FROM   EMP;
```

NỘI DUNG CHÍNH

THÁO TÁC TRÊN MỘT NHÓM

TRẦN THỊ NGỌC CHÂU

THAO TÁC TRÊN NHÓM

- CÁC HÀM DÙNG CHO NHÓM

Hàm	Giá trị trả về
AVG([DISTINCT | ALL]n)	Giá trị trung bình của n, bỏ qua trị rỗng
COUNT([DISTINCT | ALL]expr*)	Đếm số hàng của expr.
MAX([DISTINCT | ALL]expr)	Giá trị lớn nhất của expr
MIN([DISTINCT | ALL]expr)	Giá trị nhỏ nhất của expr.
SUM([DISTINCT | ALL]n)	Tổng các giá trị của n, bỏ qua trị null.

THAO TÁC TRÊN NHÓM

GROUP BY :

chia cac dòng trong thành từng nhóm nhỏ

- SELECT JOB,AVG(SAL) FRO EMP
- GROUP BY JOB;

Mệnh đề HAVING

Được dùng để mô tả những nhóm nào được hiển thị.

Ví dụ :

```
SELECT DEPTNO,MIN(SAL) FROM EMP
GROUP BY DEPTNO
HAVING COUNT (*) >3;
```

3. Sử DụNG CÔNG Cụ FORM &REPORT

From

- Các thành phần chính: canvas, window, frame, item, block, …
- Điều khiển các hành vi của form và định dạng bằng các thuộc tính, lớp thuộc tính
- Tạo menu, popup menu và item dạng phân cấp

Report

- tạo nhanh báo biểu bằng wizard
- Sử dụng các định dạng PDF, Postscript, XML, …
- Truyền tham số