Giảng viên: Ths. Nguyễn Thị Kim Phụng - Đại học CNTT

HỆ QUẢN TRỊ CSDL ORACLE

CHƯƠNG 1

KHOA HỆ THỐNG THÔNG TIN

Chương 1: Giới thiệu các công cụ SQL*Plus, iSQLPlus, OEM và Ngôn ngữ truy vấn SQL

- 1. Giới thiệu Oracle, các phiên bản.
 - 2. Công cụ SQL*Plus
 - 3. Công cụ iSQLPlus
 - 4. Công cụ EM
 - 5. Ngôn ngữ SQL (Các lệnh định nghĩa dữ liệu, thao tác dữ liệu, truy vấn dữ liệu, điều khiển dữ liệu, phân quyền users, roles).
 - 6. Các lệnh giao tác
 - 7. Sequences, Views, Indexes, Synnonym
 - 8. Oracle data dictionary

1. Giới thiệu Oracle (1)

- Oracle là tên của một hãng phần mềm, một hệ quản trị cơ sở dữ liệu phổ biến trên thế giới . Hãng Oracle ra đời đầu những năm 70 của thế kỷ 20 tại Mỹ.
- Tập hợp các sản phẩm phần mềm phục vụ cho mục đích xây dựng và quản lý hệ thống thông tin, các ứng dụng giao tiếp cơ sở dữ liệu bên dưới.
- Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mang tính mềm dẻo, linh động, thích ứng cao với các quy mô xử lý giao dịch, an toàn hệ thống. Cung cấp các công cụ xây dựng và quản lý cơ sở dữ liệu.
- Tích hợp Web: kết nối ứng dụng với công nghệ Web được tích hợp trong Oracle WebServer.

1. Giới thiệu Oracle (1) – Các phiên bản

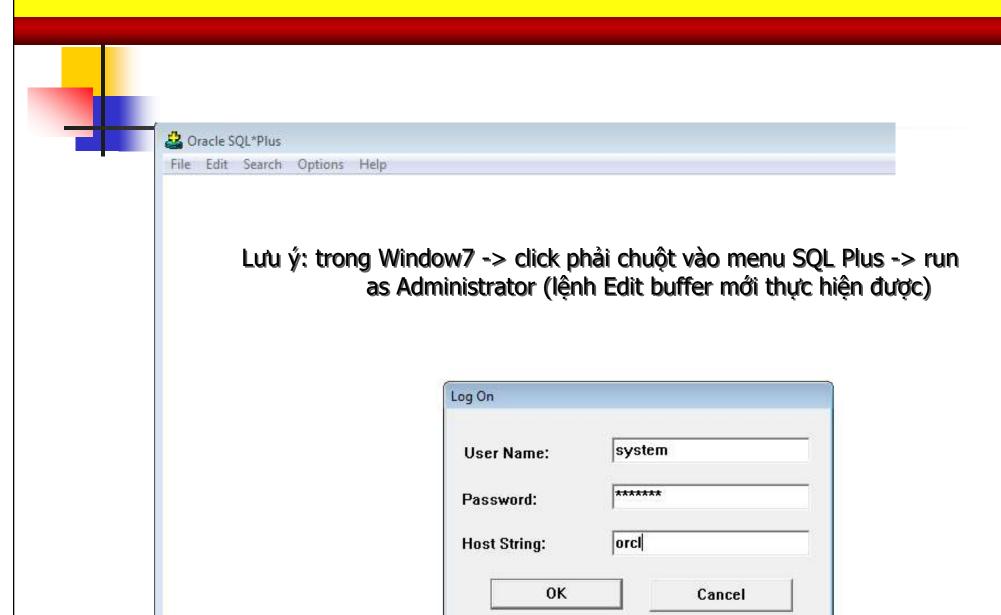
- Phiên bản 1 phát hành năm 1977.
- Phiên bản 2 phát hành năm 1979.
- Phiên bản 3 phát hành năm 1983.
- Phiên bản 4 phát hành năm 1984.
- Phiên bản 5 phát hành năm 1985 (SQLNet: hệ thống khách/chủ (client/server)).
- Phiên bản 6 phát hành năm 1988 (Sequence, thao tác ghi trễ).
- Oracle7 được phát hành năm 1992 (SQL*DBA).
- Năm 1999 Oracle giới thiệu Oracle8i (i:internet).
- Năm 2001-2002: 2 phiên bản Oracle9i (Release 1&2).
- Năm 2004-2005: 2 phiên bản Oracle10g (g:Grid) (Release 1&2).
- Năm 2007-2009: phiên bản Oracle11g (Release 1&2).
- http://www.oracle.com/technology/software/products/database/index.h tml

1. Giới thiệu Oracle (2) – Các sản phẩm



- Database Server (Server quản lý cơ sở dữ liệu)
- Công cụ thao tác cơ sở dữ liệu: SQL*Plus
- Công cụ phát triển ứng dụng: Oracle Developer Suite (Form, Report,...), Oracle JDeveloper,...
- Phân tích dữ liệu: Oracle Discoverer, Oracle Express, Oracle Warehouse Builder,...
 - Oracle Application Server (OAS)
- Úng dụng đóng gói: Oracle Human Resource, Oracle Financial Applications,...
- Oracle Email, Oracle Calendar, Oracle Web Conferencing,...

2. Công cụ SQL*Plus (1) – Giao diện



2. Công cụ SQL*Plus (2) – Giao diện

03-MAR-10

SQL>



7

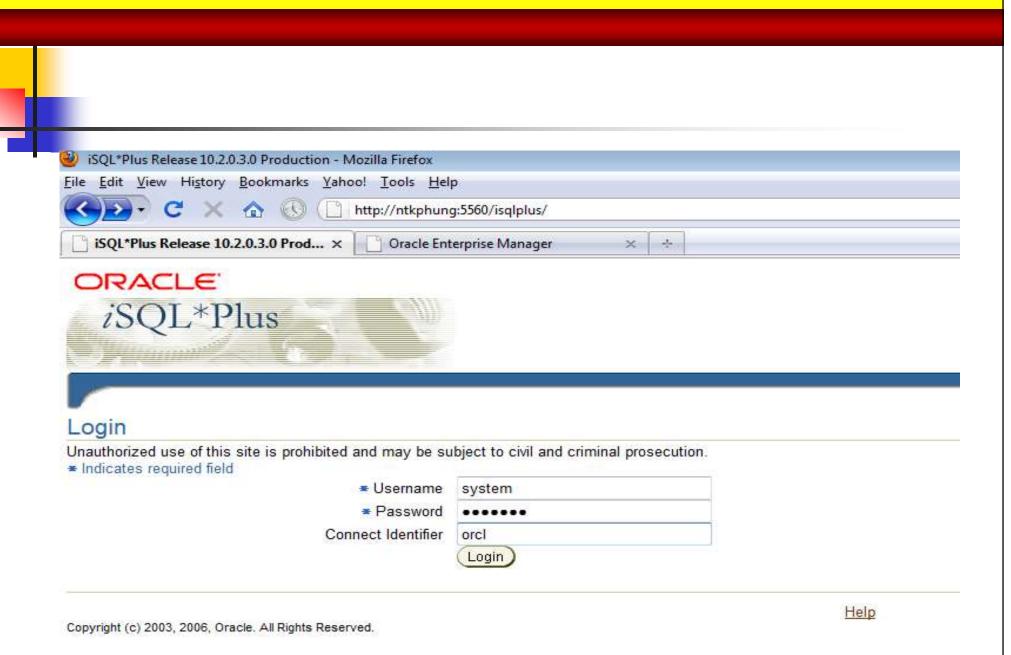
2. Công cụ SQL*Plus (3)

- Ngôn ngữ SQL: ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ, dùng trong Oralce khi cần truy xuất CSDL.
- Ngôn ngữ PL/SQL: ngôn ngữ thủ tục của Oracle dùng để xây dựng các ứng dụng, kết hợp SQL để truy xuất dữ liệu.
- Công cụ SQL*Plus: sản phẩm của Oracle, là môi trường để thực hiện các lệnh SQL và PL/SQL.
- SQL*Plus có các lệnh sau để điều khiển cách xử lý của SQL*Plus, định dạng dữ liệu xuất.

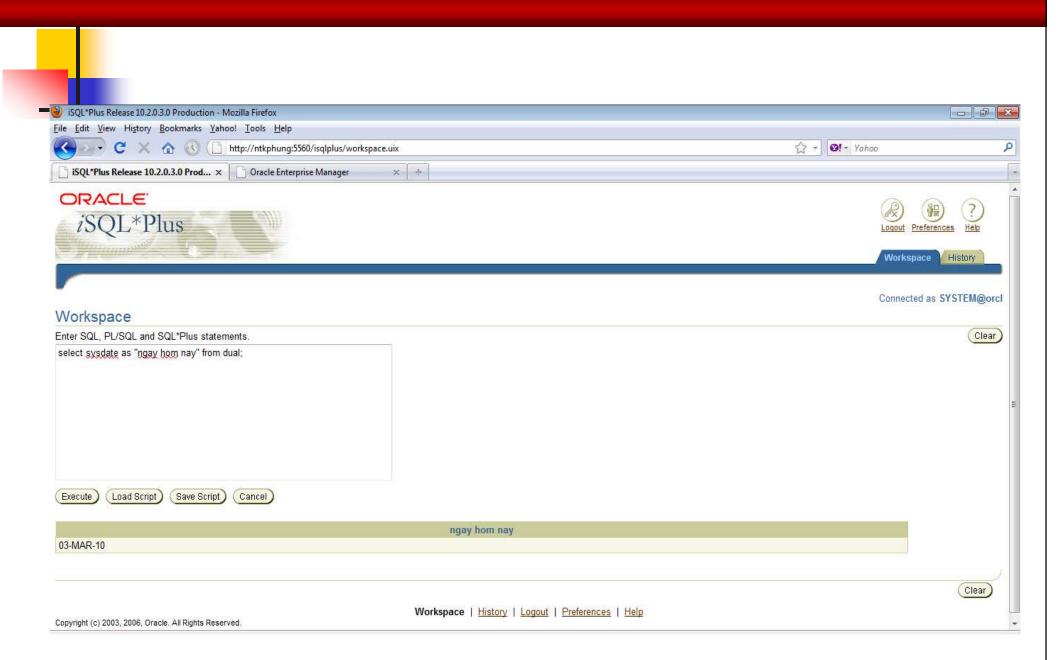
2. Công cụ SQL*Plus (4) – Các lệnh thường dùng

Lệnh	Mục đích	
CLEAR BUFFER	Xóa tất cả dòng lệnh từ SQL Buffer (file "afdeit.buf")	
DEL	Xóa dòng lệnh hiện hành trong Buffer	
LIST	Liệt kê tất cả các hàng trong SQL Buffer	
LIST n	Liệt kê hàng thứ n trong SQL Buffer	
LIST m n	Liệt kê các hàng trong phạm vi từ m đến n	
RUN (chạy trong Buffer (file "afdeit.buf"))	Hiển thị câu SQL và thực thi lệnh SQL hiện hành	
/	Chỉ thực thi lệnh SQL hiện hành trong Buffer, ko in ra	
SAVE filename	Lưu nội dung hiện hành của SQL Buffer vào filename	
GET filename	Xuất nội dung của filename	
START filename	Thực thi lệnh trong filename	
@filename	Thực thi lệnh trong filename	
EDIT	Soạn thảo nội dung file Buffer (file "afdeit.buf")	
EDIT filename	Soạn thảo nội dung filename	
EXIT	Thoát khỏi môi trường SQL*Plus	
CONNECT <u>user/password@service_name</u>	Dùng Username và password connect đến DB Server	

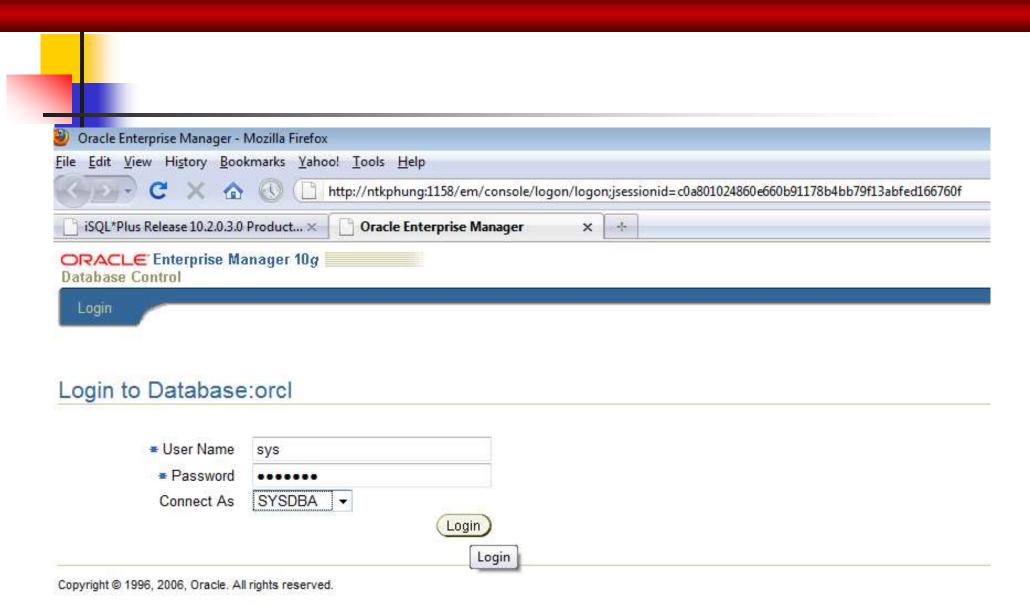
3. Công cụ iSQLPlus (1) – Thực thi lệnh SQL,PL/SQL trên web



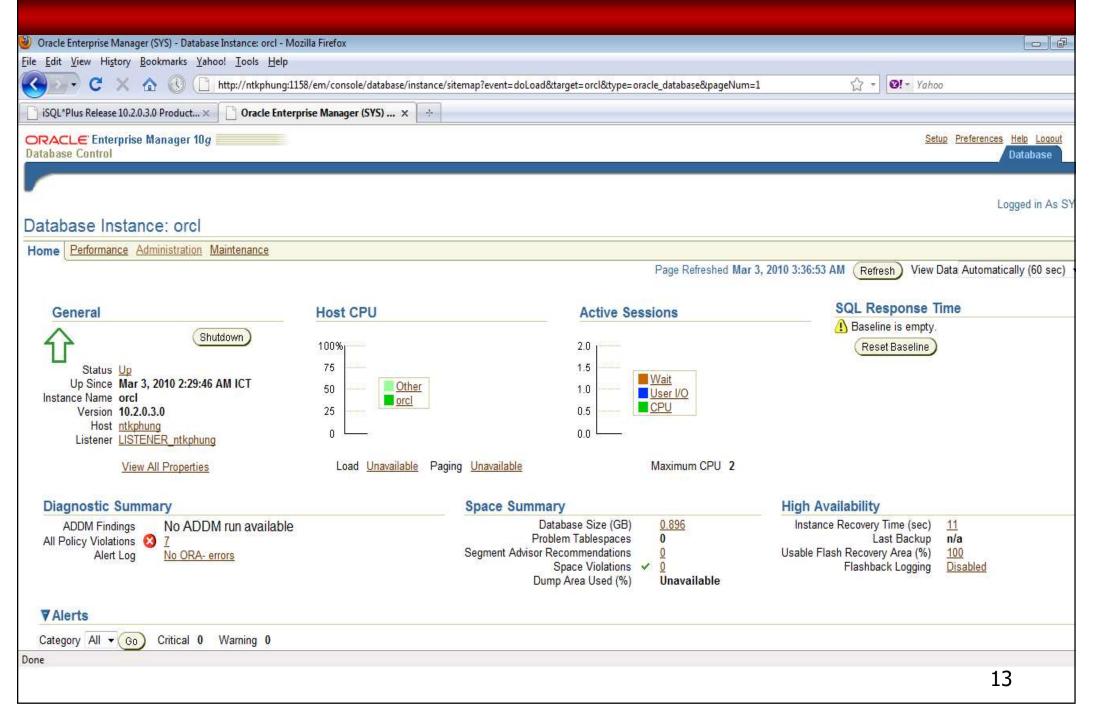
3. Công cụ iSQLPlus (2) – Thực thi lệnh SQL,PL/SQL trên web



4. Công cụ Oracle Enterprise Manager - (OEM) quản trị Oracle qua web (1)



4. Công cụ Oracle Enterprise Manager - (OEM) quản trị Oracle qua web (2)



SỬ DỤNG BIỂN THAY THẾ &, &&

- Các loại biến trong SQL*Plus: 2 kiểu biến
- Biến thay thế &: dấu & đặt trước biến. Biến được nhập giá trị lúc thực thi câu SQL. Kết quả câu SQL tùy thuộc vào giá trị nhập cho biến.
- Ví dụ: SELECT MaNV, HoNV, TenNV

FROM NhanVien

WHERE MaPhong=&DEPT_NUMBER;

Khi chạy lệnh SQL trong môi trường SQL*Plus sẽ hiện ra dòng chữ -> nhập giá trị vào (VD:5).

SQL>Enter value for DEPT_NUMBER:5

- Lưu ý: biến kiểu chuỗi, kiểu ngày đặt trong cặp dấu \'

SỬ DỤNG BIẾN THAY THẾ &, &&

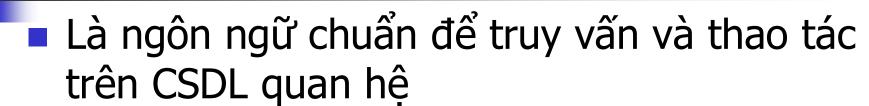
- Biến thay thế &&: dấu && đặt trước biến. Giá trị nhập một lần và được lưu trữ cho những lần sau (không nhập lại, chương trình không hỏi lại mà tự nhớ giá trị cho lần chạy sau).
- Ví dụ: SELECT MaNV, HoNV, TenNV
 FROM NhanVien
 WHERE MaPhong=&&DEPTNO;

Khi chạy lệnh SQL trong môi trường SQL*Plus sẽ hiện ra dòng chữ -> nhập giá trị vào (VD:5).

SQL>Enter value for DEPTNO:5

Ghi chú: Lần sau chạy câu lệnh, giá trị này được nhợ.

5. Ngôn ngữ SQL (1) – Giới thiệu



- Là ngôn ngữ phi thủ tục
- Khởi nguồn của SQL là SEQUEL Structured English Query Language, năm 1974)
- Các chuẩn SQL
 - SQL89 (SQL1)
 - SQL92 (SQL2)
 - SQL99 (SQL3)

5. Ngôn ngữ SQL (2) – Phân loại

- Ngôn ngữ định nghĩa dữ liệu (DDL): tạo table (bảng), view (khung nhìn), sửa cấu trúc table và thêm, xóa RBTV, xóa table, xóa view, đổi tên table
- Ngôn ngữ thao tác dữ liệu (DML): thêm, xóa, sửa dữ liệu, và truy vấn dữ liệu.
- Ngôn ngữ điều khiển dữ liệu (DCL): tạo quyền hạn, xóa quyền, làm cho hiệu lực/mất hiệu lực quyền, tạo người dùng, đổi mật khẩu, xóa người dùng, cấp quyền và thu hồi quyền sử dụng trên cơ sở dữ liệu.
- Ngoài ra còn có các lệnh điều khiển giao tác.
- Lệnh thao tác trên các thành phần CSDL khác: Synonym, Index và Sequence

5. Ngôn ngữ SQL (3) – Ngôn ngữ định nghĩa dữ liệu



- Ngôn ngữ định nghĩa dữ liệu (DDL– Data Definition Language)
- Bao gồm:
 - Lệnh tạo table, tạo view (CREATE...)
 - Lệnh sửa cấu trúc table, thêm, xóa ràng buộc toàn vẹn trên table (ALTER...)
 - Lệnh xóa table, xóa view (DROP...)
 - Đối tên table (RENAME...)

5. Ngôn ngữ SQL (4) – Ngôn ngữ thao tác dữ liệu



- Ngôn ngữ thao tác dữ liệu (DML Data Manipulation Language)
- Bao gồm:
 - Lệnh thêm dữ liệu (INSERT...)
 - Lệnh sửa dữ liệu (UPDATE...)
 - Lệnh xóa dữ liệu (DELETE...)
 - Truy vấn dữ liệu (SELECT...)

5. Ngôn ngữ SQL (5) – Ngôn ngữ điều khiển dữ liệu

- Ngôn ngữ điều khiển dữ liệu (DCL Data Control Language). Bao gồm:
 - Lệnh tạo quyền hạn (Create Role...)
 - Lệnh thiết lập, đổi hay bỏ mật khẩu của role (Alter Role...)
 - Lệnh xóa quyền hạn (Drop Role...)
 - Lệnh tạo người dùng, đổi mật khẩu và xóa người dùng (Create User..., Alter User..., Drop User...)
 - Lệnh cấp quyền cho người sử dụng cơ sở dữ liệu (GRANT...)
 - Lệnh thu hồi quyền hạn của người sử dụng cơ sở dữ liệu (REVOKE...)

5. Lệnh điều khiển giao tác + các đối tượng khác

- Lệnh điều khiển giao tác bao gồm:
 - Lệnh COMMIT, Iệnh ROLLBACK, Iệnh SAVEPOINT, Iênh AUTOCOMMIT
- Các đối tượng khác:
 - SYNONYM: tạo một Synonym (Create Synonym...),
 xóa Synnonym (Drop Synonym...)
 - INDEX: tạo chỉ mục cho table (Create Index...), bảo đảm giá trị duy nhất trong cột, thường là giá trị Primary key.
 - SEQUENCE: tạo giá trị SEQUENCE cho cột (Create Sequence...).



Cú pháp CREATE TABLE < tên table > tên côt1 kiểu dữ liêu [not null], tên côt2 kiểu dữ liêu [not null], tên_côtn kiểu_dữ_liệu [not null], khai báo khóa chính, khóa ngoại, ràng buộc toàn ven



Ngôn ngữ định nghĩa dữ liệu Tạo table, view (2)

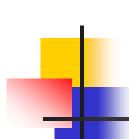
Cú pháp
 CREATE [OR REPLACE] [FORCE] VIEW
 <tên_view> [tên-cột1, cột2,...]
 AS
 SELECT ...
 [Điều kiện] [Ràng buộc]

- Ghi chú: tùy chọn Replace sẽ xóa view và tạo view mới nếu view đã tồn tại rồi.

Ngôn ngữ định nghĩa dữ liệu

Tạo table, view - Kiểu dữ liệu (3)

Loại dữ liệu	Mô tả
VARCHAR2(n)	Dữ liệu kiểu ký tự, n<=4000
CHAR(n)	Dữ liệu kiểu ký tự, kích thước cố đinh, n<=2000
NUMBER	Kiểu số nguyên, số ký số tối đa là 38 ký số
NUMBER(p)	Kiểu số nguyên, với số ký số tối đa là p
NUMBER(p,s)	Kiểu số thực, tối đa p ký số, s số thập phân. p≤38, -84 ≤ s ≤ 127.Ví dụ: số 7456123, khai báo kiểu number (7, -2) = 7456100
DATE	Kiểu ngày, lưu ngày từ 1/1/4712 BC -> 31/12/9999
LONG	Kiểu ký tự
RAW	Chuỗi nhị phân dài tối đa 2000 bytes
LONG RAW	Chuỗi nhị phân dài tối đa 2GB
BLOB	(Bynary Large Object) có độ dài ≤ 4GB
CLOB	(Character Large Object) có độ dài ≤ 4GB
BFILE	Chứa con trỏ chỉ đến một tập tin nhị phân ở ngoài DB 24



Ngôn ngữ định nghĩa dữ liệu Tạo table, view (4)

Cho lược đồ CSDL "quản lý đề án cty" như sau NHANVIEN (MaNV, HoNV, TenLot, TenNV, Phai, Luong, Phong, NgaySinh, DiaChi, Ma_NQL) PHONGBAN (MaPHG, TenPHG, TrPHG, NG NhanChuc) DEAN (MaDA, TenDA, DDIEM_DA, Phong) PHANCONG (MaNV, MaDA, ThoiGian) **DIADIEM_PHG (MaPHG, DIADIEM)** THANNHAN (MaNV, TenTN, Phai, NGSinh, QuanHe)

MANV	HOTEN	NTNS	PHAI	MA_NQL	PHONG	LUONG
001	Vuong Ngoc Quyen	22/10/1957	Nu		QL	3.000.000
002	Nguyen Thanh Tung	09/01/1955	Nam	001	NC	2.500.000
003	Le Thi Nhan	18/12/1960	Nu	001	DH	2.500.000
004	Dinh Ba Tien	09/01/1968	Nam	002	NC	2.200.000
005	Bui Thuy Vu	19/07/1972	Nam	003	DH	2.200.000
006	Nguyen Manh Hung	15/09/1973	Nam	002	NC	2.000.000
007	Tran Thanh Tam	31/07/1975	Nu	002	NC	2.200.000
008	Tran Hong Minh	04/07/1976	Nu	004	NC	1.800.000

NHANVIEN

PHANCONG

DEAN

MADA	TENDA	PHONG	NamThucHien
TH001	Tin hoc hoa 1	NC	2002
TH002	Tin hoc hoa 2	NC	2003
DT001	Dao tao 1	DH	2004
DT002	Dao tao 2	DH	2004

PHONGBAN

MAPHG	TENPH	TRPH
QL	Quan Ly	001
DH	Dieu Hanh	003
NC	Nghien Cuu	002

MANV	MADA	THOIGIAN
001	TH001	30,0
001	TH002	12,5
002	TH001	10,0
002	TH002	10,0
002	DT001	10,0
002	DT002	10,0
003	TH001	37,5
004	DT001	22,5
004	DT002	10,0
006	DT001	30,5
007	TH001	20,0
007	TH002	10,0
008	DT002	12,5

Ngôn ngữ định nghĩa dữ liệu Tạo table, view (5)

Ví dụ: câu lệnh để tạo một table nhân viên CREATE TABLE NHANVIEN

```
MANV varchar2(10) NOT NULL,
HONV varchar2(50) NOT NULL, TENLOT varchar2(50) NOT NULL,
TENNV varchar2(50) NOT NULL, NGAYSINH date,
PHAI varchar2(3) NOT NULL, DIACHI varchar2(100),
MA_NQL varchar2(10),
PHONG varchar2(10),
LUONG number,
CONSTRAINT PK_NV PRIMARY KEY (MANV),
CONSTRAINT FK_NV_PB FOREIGN KEY (PHONG) REFERENCES
PHONGBAN (MAPHG)
```



Ví dụ: câu lệnh để tạo một view chứa họ tên nhân viên phòng số 5 và tên phòng ban họ trực thuộc

CREATE VIEW TrucThuoc

AS SELECT MANV, HONV, TENNV, TENPHG FROM NhanVien nv, PhongBan p WHERE nv.PHONG = p.MAPHG AND p.nv.PHONG = 5



- Thêm thuộc tính
 - `ALTER TABLE tên_table ADD tên_cột kiểu_dữ_liệu
 - Ví dụ: thêm cột Ghi_chú vào table nhân viên ALTER TABLE NHANVIEN ADD GHI_CHU varchar2(20)
- Sửa kiểu dữ liệu thuộc tính
 ALTER TABLE tên_table MODIFY tên_cột kiểu dữ liêu mới



Ngôn ngữ định nghĩa dữ liệu Sửa cấu trúc table (8)

Ví dụ: sửa kiểu dữ liệu của cột Ghi chú

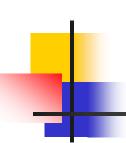
ALTER TABLE NHANVIEN **MODIFY** GHI_CHU varchar2(30)

Xóa thuộc tính

ALTER TABLE tên_bảng DROP COLUMN tên_cột

Ví dụ: xóa cột Ghi_chú từ bảng nhân viên

ALTER TABLE NHANVIEN DROP COLUMN GHI_CHU



Ngôn ngữ định nghĩa dữ liệu Sửa cấu trúc table (9)

Thêm ràng buôc toàn ven

UNIQUE (tên cột) PRIMARY KEY (têncột1,2,.) **ALTER TABLE <ten bang> ADD CONSTRAINT** FOREIGN KEY (tên cột) <ten_ràng buộc> REFERENCES tên bảng (cột_là_khóa_chính) CHECK (tên_cột điều kiên)

31

Ngôn ngữ định nghĩa dữ liệu Sửa cấu trúc bảng (10)

- Ví dụ
 - ALTER TABLE NHANVIEN ADD CONSTRAINT PK_NV PRIMARY KEY (MANV)
 - ALTER TABLE NHANVIEN ADD CONSTRAINT FK_NV_PB FOREIGN KEY (PHONG) REFERENCES PHONGBAN(MAPHG)
 - ALTER TABLE NHANVIEN ADD CONSTRAINT CHK CHECK (PHAI IN ('Nam') OR ('Nu'))
 - ALTER TABLE NHANKHAU ADD CONSTRAINT UQ_NK UNIQUE (CMND)

Ngôn ngữ định nghĩa dữ liệu Sửa cấu trúc bảng (11)

 Xóa ràng buộc toàn vẹn
 ALTER TABLE tên_bảng DROP CONSTRAINT tên_ràng_buộc

Ví dụ
 ALTER TABLE NHANVIEN DROP CONSTRAINT FK NV PB

Ngôn ngữ định nghĩa dữ liệu Xóa table, xóa view (12)

- Cú pháp xóa table DROP TABLE tên_table
- Cú pháp xóa view DROP VIEW tên_view
 - Ví dụ: xóa bảng (table) nhân viên DROP TABLE NHANVIEN
 - Ví dụ: xóa khung nhìn (view) TrucThuoc
 DROP VIEW TrucThuoc



Cú pháp

```
INSERT INTO tên_bảng VALUES (giá_tri_1, giá_tri_2,..., giá_tri_n)
INSERT INTO tên_bảng (cột1, cột2) VALUES (giá_tri_1, giá_tri_2)
```

To_Date('31/01/1977','dd/mm/yyyy'), '450 Trung Vuong, Ha Hoi', 'QL')

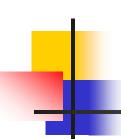


Cú pháp

```
UPDATE tên_bảng SET cột_1 = giá_tri_1, cột_2 = giá_tri_2 [WHERE điều_kiện]
```

 Ví dụ: Sửa họ nhân viên có mã số '001' thành 'Nguyen'

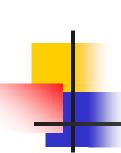
```
UPDATE NHANVIEN SET HONV = 'Nguyen' WHERE MANV='001'
```



Ngôn ngữ thao tác dữ liệu Sửa dữ liệu của bảng (3)

- Cú pháp
 - Ví dụ: Sửa họ tên của nhân viên có mã số '001' thành 'Nguyen Thanh Tung' và ngày sinh mới là 1/1/1978

```
UPDATE NHANVIEN SET HONV = 'Nguyen',
TENLOT = 'Thanh', TENNV = Tung',
NGAYSINH=To_date('1/1/1978',
'dd/mm/yyyy') WHERE MANV='001'
```



Ngôn ngữ thao tác dữ liệu Xóa dữ liệu trong bảng (4)

- Cú pháp
 DELETE FROM tên_bảng [WHERE điều_kiện]
 - Ví dụ: xóa nhân viên có mã số '001'
 DELETE FROM NHANVIEN
 WHERE MANV='001'
 - Ví dụ: xóa toàn bộ nhân viên
 DELETE FROM NHANVIEN



Câu truy vấn tổng quát SELECT [DISTINCT] tên_côt | hàm FROM bảng [WHERE điều kiên] [GROUP BY côt] [HAVING điều kiên] [ORDER BY côt ASC | DESC]

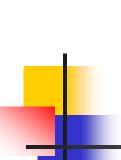


Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (6)

■ Toán tử so sánh

```
=
>
<
>=
<
>=
<>
```

- Toán tử logic: AND, OR, NOT
- Phép toán: +, ,* , /

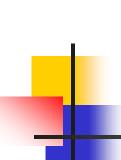


Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (7)

- Các toán tử so sánh khác
 - BETWEEN định nghĩa một đoạn giá trị liên tục
 - IS NULL kiểm tra giá trị thuộc tính có null hay không
 - LIKE kiểm tra chuỗi ký tự tương tự
 - IN kiểm tra giá trị thuộc tính có thuộc tập hợp các giá trị đã định nghĩa hay không
 - EXISTS mang giá trị TRUE nếu mệnh đề so sánh trả về ít nhất một bộ (record), FALSE nếu ngược lại

Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (8)

- Toán tử BETWEEN
 - Ví dụ: Tìm nhân viên sinh vào khoảng 1965 và 1977. Tìm nhân viên có lương không nằm trong khoảng 100000 đến 300000
 - SELECT * FROM NHANVIEN WHERE To_number(To_char(NGAYSINH,'yyyy')) BETWEEN 1965 AND 1977
 - SELECT n.MANV, n.TENNV, p.TENPHG
 FROM NHANVIEN n, PHONGBAN p
 WHERE n.PHONG=p.MAPHG
 AND n.LUONG NOT BETWEEN 100000 AND 300000



Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (9)

- Toán tử IS NULL
 - Ví dụ: tìm những nhân viên có người quản lý
 - a. SELECT * FROM NHANVIEN

WHERE MA_NQL IS NOT NULL

Ví dụ: tìm những nhân viên không có người quản lý

b. SELECT * FROM NHANVIEN

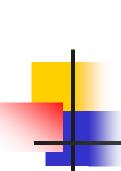
WHERE MA_NQL IS NULL



Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (10)

- Toán tử LIKE
 - So sánh chuỗi tương đối
 - Cú pháp: s LIKE p, p có thể chứa % hoặc _
 - %: thay thế một chuỗi ký tự bất kỳ
 - _ : thay thế một ký tự bất kỳ
 - Ví du SELECT * FROM NHANVIEN

WHERE HONV LIKE 'Nguyen%'



Ngôn ngữ thao tác dữ liệu Toán tử truy vấn (11)

- Toán tử IN
 - Ví dụ
 - a. SELECT * FROM NHANVIEN
 WHERE PHONG IN ('NC','QL','DH')
 - b. SELECT MANV, TENNV, DIACHI

FROM NHANVIEN

WHERE MANV NOT IN (SELECT MANV FROM PHANCONG)



Ngôn ngữ truy vấn dữ liệu Toán tử truy vấn (12)

- Toán tử EXISTS
 - Ví dụ: tìm nhân viên làm việc cho tất cả các đề án

SELECT * FROM NHANVIEN n WHERE NOT EXISTS

(SELECT * FROM DEAN d WHERE NOT EXISTS

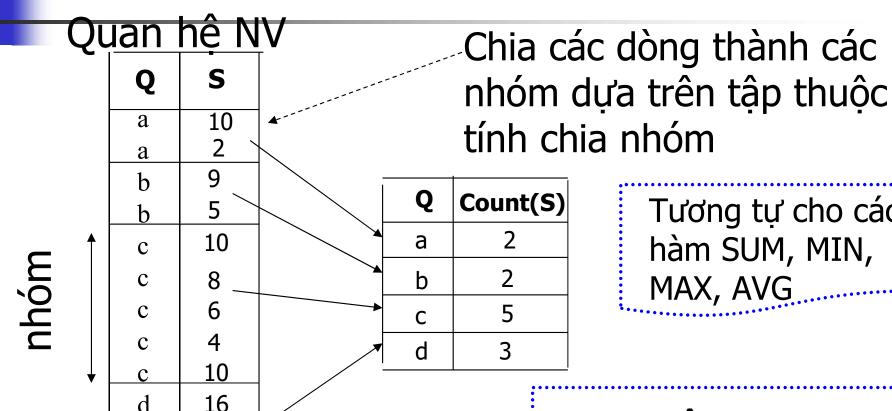
(SELECT * FROM PHANCONG p WHERE n.MANV=p.MANV AND d.MADA=p.MADA))



Mệnh đề GROUP BY

- Chia các dòng thành các nhóm nhỏ dựa trên tập thuộc tính chia nhóm.
- Tất cả các thành viên của nhóm đều thỏa các thuộc tính này.
- Thực hiện các phép toán trên nhóm như: Count (thực hiện phép đếm), Sum (tính tổng), Min(lấy giá trị nhỏ nhất), Max(lấy giá trị lớn nhất), AVG (lấy giá trị trung bình).

Ngôn ngữ truy vấn dữ liệu Mệnh đề GROUP BY (14)



Tương tư cho các hàm SUM, MIN, MAX, AVG.....

Các thuộc tính GROUP BY: Q

18

50

Câu SQL:

Select Q, count(S) From NV



Ngôn ngữ truy vấn dữ liệu Mệnh đề GROUP BY (15)

Các hàm SQL cơ bản

- COUNT: Đếm số bộ dữ liệu của thuộc tính
- MIN: Tính giá trị nhỏ nhất
- MAX: Tính giá trị lớn nhất
- AVG: Tính giá trị trung bình
- SUM: Tính tổng giá trị các bộ dữ liệu

Ngôn ngữ truy vấn dữ liệu Mệnh đề GROUP BY (16)

- Ví dụ: tìm tổng lương, lương lớn nhất, lương ít nhất và lương trung bình của các nhân viên
 - SELECT SUM(LUONG), MAX(LUONG), MIN(LUONG), AVG(LUONG) FROM NHANVIEN;
- Ví dụ: tìm tổng lương, lương lớn nhất, lương ít nhất và lương trung bình của các nhân viên phòng "Nghiên cứu"
 - SELECT SUM(LUONG), MAX(LUONG), MIN(LUONG), AVG(LUONG) from NHANVIEN, PHONGBAN WHERE MAPHG=PHONG AND TENPHG='Nghien cuu';
- Ví dụ: cho biết số lượng nhân viên SELECT COUNT(*) FROM NHANVIEN;



Ngôn ngữ truy vấn dữ liệu Mệnh đề GROUP BY (17)

■ Ví dụ

```
SELECT n.MANV, n.TENNV, n.PHONG,
MIN(p.THOIGIAN) thap_nhat,
MAX (p.THOIGIAN) cao_nhat, AVG(p.THOIGIAN)
trung_binh,
SUM (p.THOIGIAN) tong_so_gio
FROM NHANVIEN n, PHANCONG p
WHERE n.MANV=p.MANV
GROUP BY n.MANV, n.TENNV, n.PHONG
```



Ngôn ngữ truy vấn dữ liệu Mệnh đề HAVING (18)

- Mệnh đề HAVING
 - Lọc kết quả theo điều kiện, sau khi đã gom nhóm
 - Điều kiện của HAVING là điều kiện các hàm tính toán trên nhóm (Count, Sum, Min, Max, AVG) và các thuộc tính trong danh sách GROUP BY.



Ngôn ngữ truy vấn dữ liệu Mệnh đề HAVING (19)

Ví dụ: cho biết tên từng phòng ban và tổng số nhân viên, mức lương trung bình của phòng trên 2000000

SELECT p.TENPHG, COUNT(*) so_luong_nv,

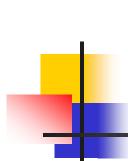
AVG(n.LUONG) luong_tb

FROM NHANVIEN n, PHONGBAN p

WHERE n.PHONG = p.MAPHG

GROUP BY p.TENPHG

HAVING AVG(n.LUONG) > 2000000



Ngôn ngữ truy vấn dữ liệu Mệnh đề HAVING (20)

- Ví dụ
 - Liệt kê nhân viên có số giờ làm việc nhiều nhất trong công ty.

SELECT n.MANV, n.TENNV, SUM (p.THOIGIAN)

FROM NHANVIEN n, PHANCONG p

WHERE n.MANV = p.MANV

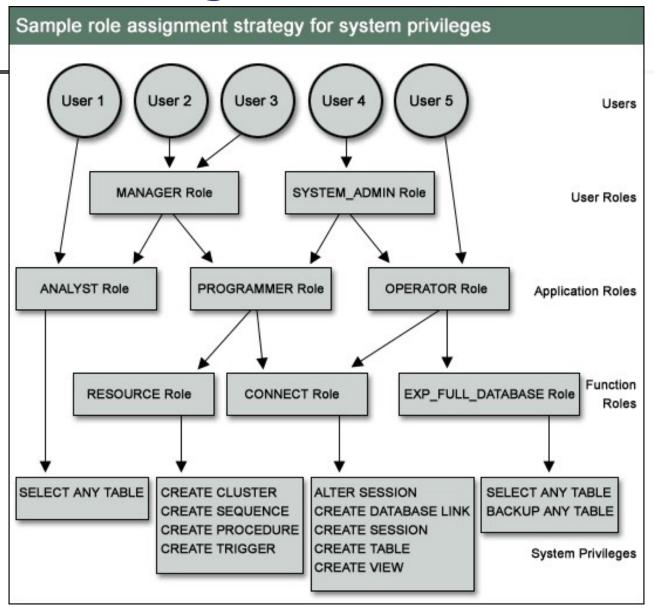
GROUP BY n.MANV, n.HOTEN

HAVING SUM (p.THOIGIAN) >= ALL (SELECT SUM(THOIGIAN) FROM PHANCONG GROUP BY (MANV)

Ngôn ngữ điều khiển dữ liệu (DCL) Giới thiệu System và Object Privileges (1)

- Ngôn ngữ điều khiển dữ liệu Data Control Language (DCL) được dùng để cấp phát, thu hồi quyền trên các đối tượng như: tables, views, sequences, synonyms, procedures đến các user và role.
- Lệnh DCL bao gồm 2 lệnh:
 - GRANT : Use to **grant privileges on tables, view, procedure** to other users or roles.
 - REVOKE :Use to take back privileges granted to other users and roles.
- Phân loại quyền: có 2 loại
 - System Privileges
 - Object privileges
- System Privileges thường được cấp bởi DBA đến các users. Ví dụ một số quyền hệ thống: CREATE SESSION, CREATE TABLE, CREATE USER,.... System privileges grants cover many areas of access in a broad brush, với những quyền tương tự như select any table. Ví dụ: grant create any cluster to customer_role; grant select any table to fred; grant create tablespace to dba_role;
- Object privileges là các quyền Insert, Select, Update, Delete, Alter, index, Refrences, All trên các object như: tables, views, synonyms, procedure. Những quyền này được cấp bởi người sở hữu object.

Ngôn ngữ điều khiển dữ liệu (2) System Privileges



Ngôn ngữ điều khiển dữ liệu (3) Object Privileges

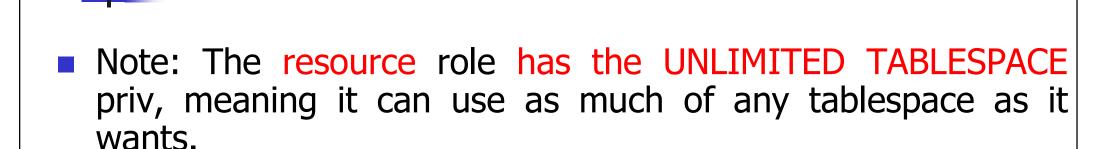
Object Privileges are

ALTER	Change the table definition with the ALTER TABLE statement.
DELETE	Remove rows from the table with the DELETE statement. Note: You must grant the SELECT privilege on the table along with the DELETE privilege.
INDEX	Create an index on the table with the CREATE INDEX statement.
INSERT	Add new rows to the table with the INSERT statement.
REFERENCES	Create a constraint that refers to the table. You cannot grant this privilege to a role.
SELECT	Query the table with the SELECT statement.
UPDATE	Change data in the table with the UPDATE statement.
	Note: You must grant the SELECT privilege on the table along with the UPDATE privilege.

Ngôn ngữ điều khiển dữ liệu (4)

- Một User muốn tạo được schema riêng cho mình cần phải có 2 nhóm quyên hệ thống thích hợp là CONNECT đến cơ sở dữ liệu và quyên RESOURCE (tài nguyên).
- System Privileges: có các quyền (role) có thể gán cho User như: CREATE ANY TABLE, CREATE ANY VIEW,...
- Object Privileges: có 8 quyền có thể gán cho User và Role, 8 quyền bao gồm: SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX, REFERENCES, ALL.
- Role: là tên của một hay một tập hợp các quyền hệ thống nhằm quản lý các quyền cho các ứng dụng cơ sở dữ liệu hoặc nhóm người dùng (User Group).
- Một User có thể truy xuất đến nhiều Role và ngược lại.
- Có vài Role được định nghĩa trước như DBA chứa tất cả các quyền của hệ thống, hoặc CONNECT, RESOURCE.... 58

Ngôn ngữ điều khiển dữ liệu (5)



- By default, when a new user has been created, it has a 0k quota on tablespaces. Without resource (which has unlimited tablespace) this user cannot allocate any storage.
- Recommend "alter user USERNAME quota unlimited on <name-tablespace>" or "Grant UNLIMITED TABLESPACE to user"

Ngôn ngữ điều khiển dữ liệu (6) Tạo người dùng

Cú pháp

CREATE USER <user-name> NOT IDENTIFIED | IDENTIFIED BY cpassword> default tablespace USERS

Trong đó:

- NOT IDENTIFIED: không đặt password cho user.
- IDENTIFIED BY password: user phải nhập password mới có thể đăng nhập hệ thống.
- Ví dụ: tạo một user Phuong với password là P987654
 CREATE USER Phuong IDENTIFIED BY P987654

CREATE USER Phuong1 IDENTIFIED BY P987654 default tablespace TEMP

CREATE USER Phuong1 IDENTIFIED BY P987654 default tablespace USERS

Ngôn ngữ điều khiển dữ liệu (7)

Thiết lập/thay đổi/bỏ mật khẩu cho người dùng + xóa người dùng

Thiết lập/thay đổi/bỏ mật khẩu cho người dùng Cú pháp: ALTER USER <User-name> NOT IDENTIFIED | IDENTIFIED BY <password>

Ví dụ: thay đổi password mới cho user Phuong là p123 ALTER USER Phuong IDENTIFIED BY p123

Xóa người dùng

Cú pháp: DROP USER < User-name > CASCADE

Ví dụ: xóa người dùng Phuong với tất cả object của user Phuong, nếu Phuong không có object thì không cần từ khóa CASCADE

DROP USER Phuong

Ngôn ngữ điều khiển dữ liệu (8) Tạo Role (nhóm quyền)

Cú pháp

Trong đó:

- NOT IDENTIFIED: không đặt password cho role.
- IDENTIFIED BY password: user phải nhập password để làm cho role có hiệu lực.

<u>Lưu ý</u>: ký tự đầu tiên của password không được là số hoặc ký tự đặc biệt

Ví dụ: tạo một role QuanTriSV với password là QT123
 CREATE ROLE QuanTriSV IDENTIFIED BY QT123

Tạo một role TaoSynonym cho phép tạo synonym với password là TS123 CREATE ROLE TaoSynonym IDENTIFIED BY TS123

Ngôn ngữ điều khiển dữ liệu (9)

Thiết lập/thay đổi/bỏ mật khẩu cho Role + xóa Role

Thiết lập/thay đổi/bỏ mật khẩu cho quyền Cú pháp: ALTER ROLE <role-name> NOT IDENTIFIED | IDENTIFIED BY <password>

Ví dụ: đổi password mới cho role QuanTriSV là QT123456 ALTER ROLE QuanTriSV IDENTIFIED BY QT123456

Xóa quyền

Cú pháp: DROP ROLE < role-name >

Ví dụ: xóa quyền QuanTriSV

DROP ROLE QuanTriSV

Ngôn ngữ điều khiển dữ liệu (10) Cấp phát quyền

Cú pháp cấp phát quyền system Privileges cho người dùng

GRANT tên role_1, tên role_2....TO User1, User2,... [WITH GRANT OPTION]

<u>Ví du</u>: Grant **create synonym, create any table** to Phuong Grant **CONNECT, RESOURCE** to Phuong Grant **create synonym** to TaoSynonym

Cú pháp cấp phát quyền object Privileges cho người dùng

GRANT privil1, privil2,.../ALL ON <TênObject> TO User1, User2,... [WITH GRANT OPTION]

<u>Ví du</u>: Grant **select, insert** on Scott.emp TO Phuong

Ngôn ngữ điều khiển dữ liệu (11) Cấp phát quyền

Du pháp cấp phát quyền system Privileges cho Role

GRANT tên role_system_1, tên role_system_2....TO Role1, Role2,... [WITH GRANT OPTION]

<u>Ví du</u>: Grant **create synonym, create any table** to QuanTriSV Grant **CONNECT, RESOURCE** to QuanTriSV

Cú pháp cấp phát quyền object Privileges cho Role

GRANT privil1, privil2,... [ON <TênObject>] TO Role1, Role2,... [WITH GRANT OPTION]

Trong đó: **<TênObject>** : có thể là tên của một Table, View, Sequence, Synonym, Procedure, Function, Package.

privil1, privil1,...là các quyền hệ thống system Privileges (create table, create procedure, create trigger, create synonym,...) hoặc là 1 trong 8 quyền trên object Privileges để thao tác trên table hay view,...



 Cấp quyền truy xuất đến tất cả User, Role bằng lệnh GRANT privil1, privil2,.../ALL ON <TênObject>
 TO PUBLIC

Ví dụ: grant select on TênUser.TênTable to public

- Ví dụ 1: cấp phát quyền cho user Phuong GRANT INSERT, UPDATE ON TênUserTạoTableNhanVien.NHANVIEN TO Phuong
- Ví dụ 2: cấp phát quyền cho role QuanTriSV
 GRANT ALL ON TênUserTạoTableNhanVien.NHANVIEN TO QuanTriSV WITH GRANT ADMIN



■ Thu hồi quyền truy xuất từ User, Role bằng lệnh REVOKE privil1, privil2,.../ALL ON <TênObject> FROM PUBLIC

Ví dụ: revoke select on TênUser.TênTable from public

- Ví dụ 1: thu hồi quyền từ user Phuong REVOKE INSERT, UPDATE ON TênUserTạoTableNhanVien.NHANVIEN FROM Phuong
- Ví dụ 2: thu hồi quyền của role QuanTriSV REVOKE ALL ON TênUserTạoTableNhanVien.NHANVIEN FROM QuanTriSV



Ngôn ngữ điều khiển dữ liệu VD đoạn PL/SQL cấp quyền (14)

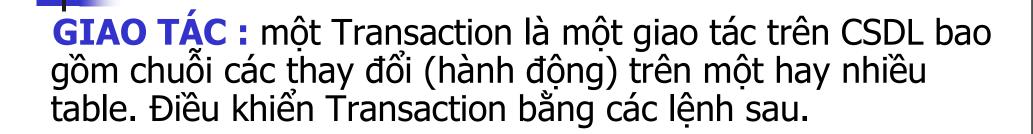
 This will grant read only access on all your objects to another schema

```
begin
for x in (select object_name, object_type from
user_objects where object_type='TABLE')
loop
execute immediate 'grant select on
'|| x.object_name || ' to &&schema';
end loop;
end;
```



- To see which table privileges are granted by you to other users.
 SELECT * FROM USER_TAB_PRIVS_MADE
- To see which table privileges are granted to you by other users SELECT * FROM USER_TAB_PRIVS_RECD;
- To see which column level privileges are granted by you to other users.
 SELECT * FROM USER_COL_PRIVS_MADE
- To see which column level privileges are granted to you by other users SELECT * FROM USER_COL_PRIVS_RECD;
- To see which privileges are granted to roles SELECT * FROM USER_ROLE_PRIVS;

6. CÁC LỆNH GIAO TÁC (1) - TRANSACTION



- Lệnh COMMIT (hoàn tất giao tác)
 - Cú pháp: COMMIT
- Lệnh ROLLBACK (phục hồi ngược lại chuỗi hành động đã thực hiện trước đó).
 - Cú pháp: ROLLBACK [TO SAVEPOINT name]
- Lệnh SavePoint
 - Cú pháp: SAVEPOINT tên-SavePoint
- Chế độ AutoCommit (AUTOCOMMIT ON/OFF)

6. CÁC LỆNH GIAO TÁC (2) - TRANSACTION

insert into student values(1,'Nguyen Van A');
savepoint A;
insert into student values(2,'Nguyen Van B');
savepoint B;
insert into student values(3,'Nguyen Van C');
savepoint C;

Ví dụ: minh họa cách dùng các lệnh SavePoint, RollBack, Commit

select * from student; -> kết quả ? sinh viên

insert into student values(4,'Nguyen Van D');

insert into student values(5,'Nguyen Van E');

savepoint D;

end;

savepoint A; /* savepoint A trước đó ko đc hiểu nữa */

- rollback to savepoint A; //thải hồi lệnh ngược đến savepoint A
- select * from student; -> kết quả ? sinh viên

6. CÁC LỆNH GIAO TÁC (3) - TRANSACTION



1) Giá trị nào được thêm vào bảng NUMBERS? Test trường hợp khác sử dụng lệnh drop table, Iệnh DROP có được Rollback??

BEGIN

FOR IX IN 9..12 LOOP

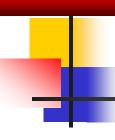
```
IF IX = 9 THEN
```

- INSERT INTO NUMBERS VALUES (IX);
- ELSIF IX = 11 THEN
- DELETE FROM NUMBERS;
- END IF;
- IF IX = 11 THEN
- ROLLBACK;
- ELSE
- COMMIT;
- END IF;

END LOOP; COMMIT;

END;

6. CÁC LỆNH GIAO TÁC (3) - TRANSACTION



2) Giá trị nào được thêm vào bảng NUMBERS?

BEGIN

FOR IX IN 9..12 LOOP

```
IF IX = 9 THEN
```

- INSERT INTO NUMBERS VALUES (IX);
- ELSIF IX = 11 THEN
- DELETE FROM NUMBERS;
- END IF;
- IF IX = 12 THEN
- ROLLBACK;
- ELSE
- COMMIT;
- END IF;

END LOOP; COMMIT;

END;

6. CÁC LỆNH GIAO TÁC (4) - TRANSACTION

ựu Ý: các lệnh liên quan cấu trúc không thể rollback, ví
 dụ: lệnh CREATE TABLE, DROP TABLE, ALTER... Khi thực
 hiện các lệnh này thì tất cả hành động trước đó đều được commit.

Ví du:

```
CREATE TABLE student (id NUMBER, name VARCHAR2(100));
INSERT INTO student VALUES (1, 'Nguyên Văn A');
CREATE TABLE course (id NUMBER, name VARCHAR2(100));
INSERT INTO course VALUES (1, 'Toan');
INSERT INTO course VALUES (2, 'Văn');
ROLLBACK;
Kết quả thực thi block trên?
```

7. CÁC ĐỐI TƯỢNG KHÁC – Synonym (1)



Synonym:

Là tên đặt cho một đối tượng cụ thể nào đó. Thường dùng synonym để tạo ra những đối tượng dùng chung.

■ Ví dụ:

Một người dùng muốn sử dụng một bảng được sở hữu bởi người dùng khác thay vì gọi "tên_người_dùng.tên_bảng" thì user đó tạo ra một synonym cho bảng đó với một tên dễ nhớ nào đó, tên thật của đối tượng được che dấu đi.

Các thao tác trên Synonym: tạo Synonym, xóa Synonym.

7. CÁC ĐỐI TƯỢNG KHÁC – Synonym (2)



■ Tạo Synonym: (user phải được cấp quyền Create Synonym trước khi sử dụng lệnh này)

Cú pháp:

CREATE SYNONYM tênSynonym **FOR** tênUser tênTable tênView Ví du:

CREATE SYNONYM nv FOR phuong.NHANVIEN;

Xóa Synonym:

Cú pháp:

DROP SYNONYM tên-synonym;

Ví du:

DROP SYNONYM nv;

7. CÁC ĐỐI TƯỢNG KHÁC – Index (1)



- Index (tạo chỉ mục): sử dụng Oracle Index nhằm
- Tăng tốc độ xây dựng lại các dòng theo một khóa đặc biệt.
- Bảo đảm giá trị duy nhất trong cột, thường là giá trị primary key.

■ Tao Index

```
<u>Cú pháp</u>: CREATE [UNIQUE] INDEX index_name ON table (column1 [, <column2>], ...)
Ví dụ:
```

CREATE UNIQUE INDEX i_cmnd ON NHAN_KHAU(SO_CMND);

7. CÁC ĐỐI TƯỢNG KHÁC – Index (2)



- Phân loại index:
- UNIQUE: Bảo đảm giá trị trong các cột là duy nhất.
 (no two rows of a table have duplicate values in the key column (or columns))
- NONUNIQUE: (là default) kết quả truy vấn có thể nhanh nhất (do not impose this restriction on the column values)
 - * Nếu là Single colum thì chỉ một cột tồn tại index.
 - * Nếu là Concatenated Index: Trên 16 cột có thể chỉ ra trong index (giới hạn 30).

7. CÁC ĐỐI TƯỢNG KHÁC – Index (3)

(concatenated index also is called Composite Index - index that you create on multiple columns in a table)

VEND ID	10_440	UNIT COST
1012	10-440 10-441 457	.25 .39 4.95
1010	10-440 457	.27 5.10
1220 1012	08-300 08-300	1.33
1292	457	5.28

CREATE INDEX VP_INDEX ON VENDOR_PARTS(VEND_ID, PART_NO);

Xóa index

<u>Cú pháp</u>: DROP INDEX index_name;

Ví du: DROP INDEX i_cmnd;

7. CÁC ĐỐI TƯỢNG KHÁC - Sequence (1)



Sequence là đối tượng tạo ra một dãy số liên tiếp một cách tự động, thường hay sử dụng trong câu lệnh INSERT để nhập dữ liệu cho bảng (ví dụ mã khách hàng tăng tự động).

Tạo Sequence

Cú pháp:

```
CREATE SEQUENCE sequence_name
[INCREMENT BY n]
[START WITH n]
[MAXVALUE n | NOMAXVALUE]
[MINVALUE n | NOMINVALUE]
[CYCLE | NOCYCLE]
[CACHE n | NOCACHE]
[ORDER | NOORDER]
```

7. CÁC ĐỐI TƯỢNG KHÁC – Sequence (2)



Ví dụ

CREATE SEQUENCE sohoadon

MINVALUE 1

MAXVALUE 9999999 //nếu không muốn giới hạn -> sử dụng NOMAXVALUE INCREMENT BY 1

START WITH 1 //nếu sử dụng lệnh ALTER thì không sử dụng với START

NOCACHE

NOORDER

NOCYCLE; //tăng đến giá trị MAXVALUE thì sequence không tăng được nữa và ngược lại cho CYCLE

Sử dụng Sequence

Giá trị hiện hành và kế tiếp của dãy sequence được lưu trong hai cột:

- Giá trị hiện hành của sequence: tên_sequence.CURRVAL (co tac dung o client, ko thay dc gia tri tang do user khac goi)
- Giá trị kế tiếp của sequence: tên_sequence.NEXTVAL (lenh co tac dung tren server)

7. CÁC ĐỐI TƯỢNG KHÁC - Sequence (3)

Tham số CACHE trong Sequence

- The Number of Values in Each Sequence Cache Entry
- When a sequence is read into the sequence cache, sequence values are generated and stored in a cache entry. These values can then be accessed quickly. The number of sequence values stored in the cache is determined by the CACHE parameter in the CREATE SEQUENCE statement. The default value for this parameter is 20.
- This CREATE SEQUENCE statement creates the seq2 sequence so that 50 values of the sequence are stored in the SEQUENCE cache:
- CREATE SEQUENCE seq2 CACHE 50; The first 50 values of seq2 can then be read from the cache. When the 51st value is accessed, the next 50 values will be read from disk.
- Choosing a high value for CACHE lets you access more successive sequence numbers with fewer reads from disk to the sequence cache. However, if there is an instance failure, then all sequence values in the cache are lost. Cached sequence numbers also could be skipped after an export and import if transactions continue to access the sequence numbers while the export is running.
- If you use the NOCACHE option in the CREATE SEQUENCE statement, then the values of the sequence are not stored in the sequence cache. In this case, every access to the sequence requires a disk read. Such disk reads slow access to the sequence. This CREATE SEQUENCE statement creates the SEQ3 sequence so that its values are never stored in the cache:
- CREATE SEQUENCE seg3 NOCACHE;

7. CÁC ĐỐI TƯỢNG KHÁC - Sequence (4)



Truy vấn giá trị hiện tại của sequence sohoadon (nếu ko gọi sohoadon.nextval một lần khi login vào session của mình trước, xem lỗi ở slide kế tiếp):

SELECT sohoadon.currval from DUAL;

Truy cập giá trị tiếp theo của sequence sohoadon bằng lệnh: SELECT sohoadon.nextval from DUAL; (DUAL là bảng tạm trong Oracle)

Sử dụng sequence trong một câu insert:

INSERT INTO HOADON (SO_HD, TongTien) VALUES (sohoadon.nextval, 5000000); /* sohoadon là sequence đã tạo trước. */

7. CÁC ĐỐI TƯỢNG KHÁC - Sequence (5)

- Lỗi truy xuất CURRVAL nếu chưa khởi tạo giá trị

- ORA-08002: sequence string.CURRVAL is not yet defined in this session
- Cause: sequence CURRVAL has been selected before sequence NEXTVAL
 Action: select NEXTVAL from the sequence before selecting CURRVAL
- Link xem các loại lỗi (Oracle Error Code Collections):
 http://www.ora-code.com/

Hai cột CURRVAL, NEXTVAL trong Oracle được gọi là **pseudo column** (**cột ảo**), người dùng chỉ được truy xuất, không được cập nhật dữ liệu trong các cột này.

Oracle có các cột ảo như: ROWID (mã dòng), ROWNUM (số thứ tự dòng), SYSDATE (ngày hiện hành của hệ thống), USER (cho biết tên user hiện hành đang đăng nhập hệ thống)...

7. CÁC ĐỐI TƯỢNG KHÁC - Sequence (6)



■ Sửa Sequence (tạo lại các thông số)

ALTER SEQUENCE tên-sequence

[INCREMENT BY n]

[MAXVALUE n | NOMAXVALUE]

[MINVALUE n | NOMINVALUE]

[CYCLE | NOCYCLE]

[CACHE n | NOCACHE]

[ORDER | NOORDER]

Xóa Sequence

Cú pháp DROP SEQUENCE tên-sequence

Ví du: DROP SEQUENCE sohoadon

8. Từ ĐIỂN DỮ LIỆU – Oracle data dictionary

MỘT SỐ VIEW TỪ ĐIỂN THƯỜNG DÙNG

- USER_objects: xem các object mà mình đã tạo
- USER_TABLES : xem các table mình đã tạo
- USER_VIEWS : xem các view mình đã tạo
- USER_INDEXES : xem các index mình tao
- USER_TYPES : xem các type mình đã tạo
- USER_SEQUENCES : xem các sequence mình tạo
- USER_PROCEDURES : xem các procedure,function mình đã tao
- USER_TRIGGERS : xem các trigger mình đã tạo
- USER_SYNONYMS : xem các synonyms mình đã tạo
- user_sys_privs: xem quyên của user hiện hành
-VD: select * from user tables

MỘT SỐ HÀM CHO NGÀY THÁNG (1) Các thao tác trên biến kiểu ngày X

Next_day(X,' Thứ m')

Hàm lấy ngày thứ **m** tiếp theo sau ngày X, ví dụ: select next_day (sysdate, 'Tuesday') from dual

Add_months(X, sốtháng)

Hàm cộng thêm n tháng sau ngày X, kết quả trả về là ngày sau n tháng của ngày X ví dụ: select Add_months (sysdate, 5) from dual

Months_between(ngày1, ngày2)

Hàm trả về số tháng giữa 2 ngày, >0 nếu ngày 1 lớn hơn ngày 2 và ngược lại ví dụ: select Months_between (sysdate, '15-June-2011') from dual

Extract ({ YEAR | MONTH | DAY | HOUR | MINUTE |
 SECOND } FROM X) - lấy ngày hoặc tháng hoặc năm của ngày X

ví dụ: select extract(YEAR FROM DATE '2003-08-22') from dual select extract(MONTH FROM sysdate) from dual select extract(DAY FROM sysdate) from dual select extract(DAY FROM cột-kiểu-ngày) from tên-bảng-chứa-cột

MỘT SỐ HÀM CHO NGÀY THÁNG (2) Các thao tác trên biến kiểu ngày X

To_Char(X,'DD'), To_Char(X,'MM'), To_Char(X, 'YYYY'), To_Char(X, 'DD-MM-YYYY')

Một loại hàm khác lấy thành phần ngày hoặc tháng hoặc năm của ngày X. ví du: select To_Char(sysdate, 'DD') from dual select To_Char(sysdate, 'MM') from dual select To_Char(sysdate, 'YYYY') from dual select To_Char(sysdate, 'DD-MM-YYY') from dual

To_Char(X, đinh-dang-ngày)

Hiển thị ngày kiểu số và có số thứ tự theo sau, dùng DD và có chữ "th" đi kèm ví dụ: select To_char (sysdate, 'DDth') from dual select To_char (sysdate, 'DDth MM') from dual select To_char (sysdate, 'DDth MM, YYYY') from dual Hiển thị ngày kiểu chữ, dùng DDsp, nếu có số thứ tự, gắn thêm "th" ví dụ: select To_char (sysdate, 'DDsp') from dual select To_char (sysdate, 'DDspth MM') from dual select To_char (sysdate, 'DDsp Month, YYYY') from dual (neu viet tat chu "Mon" sẽ hiển thị tên tháng được viết tắt, ví dụ Dec, Nov, Jun,...)

88



MỘT SỐ HÀM THƯỜNG DÙNG TRONG ORACLE

■ HỌC VIÊN XEM FILE WORD ĐI KÈM