


Hệ quản trị Cơ sở dữ liệu

Chương 2: Ngôn ngữ SQL

GV: ThS. Đỗ Thị Minh Phụng

1

1




Nội dung chi tiết

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

2

2




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.

3

3




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).
- ☐ Năm 2013: phiên bản Oracle 12c (Release 1&2).
- ☐ <http://www.oracle.com/technology/software/products/database/index.html>

4

4




Giới thiệu Oracle (1) – 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)
- Ứng dụng đóng gói: Oracle Human Resource, Oracle Financial Applications,...
- Oracle Email, Oracle Calendar, Oracle Web Conferencing,...

5

5



Công cụ SQL*Plus (1) – Giao diện

```

SQL*Plus
SQL*Plus: Release 12.2.0.1.0 Production on Mon Feb 24 18:32:47 2020
Copyright (c) 1982, 2016, Oracle. All rights reserved.
Enter user-name: SYSTEM
Enter password:
Last Successful login time: Mon Feb 24 2020 18:30:45 +07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> select sysdate as "hôm nay" from dual;

hôm nay
-----
24-FEB-20

SQL>

```

6

6

Công cụ SQL*Plus (2)

- Ngôn ngữ SQL: ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ, dùng trong Oracle 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.

7

7

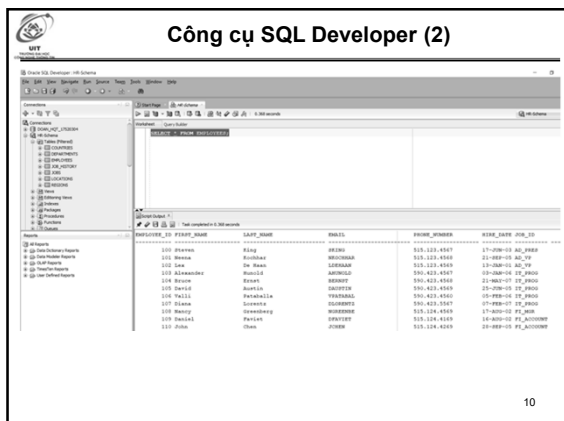
<div style="display: flex; justify-content: space-between; align-items: center;"> Công cụ SQL*Plus (3) – Các lệnh thường dùng </div>	
Lệnh	Mục đích
CLEAR BUFFER	Xóa tất cả dòng lệnh từ SQL Buffer (file "afdeit.buf")
DEL	Xóa dòng lệnh hiện hành trong Buffer
LIST	Liệt kê tất cả các hàng trong SQL Buffer
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, không 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	Sửa thảo nội dung file Buffer (file "afdeit.buf")
EDIT filename	Sửa thảo nội dung filename
EXIT	Thoát khỏi môi trường SQL*Plus
CONNECT user/password@service_name	Dùng Username và password connect đến DB Server

8

Công cụ SQL Developer (1)

Oracle SQL Developer là phần mềm do chính hãng Oracle phát triển để làm việc với Oracle Database.

9



10

Sử dụng biến 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 Phong=&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 ' '

11

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 Phong=&&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 nhớ.


12

 **Ngôn ngữ SQL (1) – Giới thiệu**

- ☐ Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ
- ☐ 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)

13


13

 **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.

14


14

 **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...)
 - Đổi tên table (RENAME...)

15

15

 **IS 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...)

16

16


 **IS 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...)

17

17

 **IS Lệnh điều khiển giao tác + các đối tượng khác**

☐ Lệnh điều khiển giao tác bao gồm:


- Lệnh COMMIT, lệnh ROLLBACK, lệnh SAVEPOINT, lệnh AUTOCOMMIT

☐ Các đối tượng khác:

- SYNONYM: tạo một Synonym (Create Synonym...), xóa Synonym (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...).


18

18

 **Ngôn ngữ định nghĩa dữ liệu**
Tạo table, view (1)

❑ **Cú pháp**
CREATE TABLE <tên_table>
(
 tên_cột_1 kiểu_dữ_liệu [not null],
 tên_cột_2 kiểu_dữ_liệu [not null],
 ...
 tên_cột_n kiểu_dữ_liệu [not null],
 khai báo khóa chính, khóa ngoại, ràng buộc toàn vẹn
)


19

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

❑ **Cú pháp**
CREATE [OR REPLACE] [FORCE] VIEW <tên_view>
[tên_cột_1, tên_cột_2,...]
AS
SELECT ...
[Điều kiện] [Ràng buộc]


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

20

 **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 định, 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	(Binary 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

21

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

Cho lược đồ CSDL “Quản lý đề án công ty” như sau

NHANVIEN (MaNV, HoNV, TenLot, TenNV, Phai, MLuong, Phong, NgSinh, DChi, Ma_NQL)

PHONGBAN (MaPHG, TenPHG, TrPHG, NGNC)

DEAN (MaDA, TenDA, DDIEM_DA, Phong, NgayBD_DK, NgayKT_DK)


PHANCONG (MaNV, MaDA, ThoiGian)

DIADIEM_PHG (MaPHG, DIADIEM)

THANNHAN (MaNV, TenTN, Phai, NGSinh, QuanHe)

22

22

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

NHANVIEN									
MANV	HONV	TENLOT	TENNV	NGSINH	PHAI	DCHI	MA_NQL	PHONG	MLUONG
001	Vương	Nam	Quang	25/10/1977	Nam	450 Truong Vuong, Ha Noi	001	QL	1.000.000
002	Nguyễn	Thanh	Tung	09/01/1955	Nam	731 Tran Hung Dao, Q1, Tp HCM	001	NC	2.500.000
003	Le	Thu	Nhan	18/12/1980	Nu	291 Ho Van Huu, Q10, Tp HCM	001	DH	2.500.000
004	Đinh	Ba	Tien	09/01/1968	Nam	638 Nguyen Van Cu, Q5, Tp HCM	002	NC	2.200.000
005	Bui	Thuy	Vu	19/07/1972	Nam	332 Nguyen Thi Hoc, Q1, Tp HCM	003	DH	2.200.000
006	Nguyễn	Minh	Hung	15/09/1973	Nam	978 Ba Ria, Vung Tam	002	NC	2.000.000
007	Tran	Thanh	Tam	31/07/1975	Nu	543 Ma Thi Loan, Q1, Tp HCM	002	NC	2.200.000
008	Tran	Hong	Van	04/07/1976	Nu	980 Le Hong Phong, Q10, Tp HCM	004	NC	1.800.000

DEAN					
MADA	TENDA	DDIEM_DA	PHONG	NGAYBD_DK	NGAYKT_DK
TH001	Tin học học 1	HANOI	NC	01/02/2003	01/02/2004
TH002	Tin học học 2	TPHCM	NC	04/06/2003	01/02/2004
DT001	Đào tạo 1	NHATRANG	DH	01/02/2002	01/02/2006
DT002	Đào tạo 2	HANOI	DH	01/02/2002	01/02/2006


PHANCONG			
MANV	MADA	THOIGIAN	
001	TH001	10.0	
001	TH002	12.5	
002	TH001	10.0	
002	TH002	10.0	
002	DT001	10.0	
002	DT002	10.0	
003	TH001	17.5	
004	DT001	22.5	
004	DT002	10.0	
006	DT001	40.5	
007	TH001	20.0	
007	TH002	10.0	
008	TH001	10.0	
008	DT002	12.5	

PHONGBAN			
MAPHG	TENPHG	TRPHG	NGNC
QL	Quan Ly	001	22/05/2000
DH	Định Hình	003	10/10/2002
NC	Ngành Cơ	002	15/03/2002

DIADIEM_PHG	
MAPHG	DIADIEM
NC	HANOI
NC	TPHCM
QL	TPHCM
DH	HANOI
DH	TPHCM
DH	NHATRANG

23

23


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

□ Ví dụ: câu lệnh để tạo 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,
  NGSINH date,
  PHAI varchar2(3) NOT NULL,
  DCHI varchar2(100),
  MA_NQL varchar2(10),
  PHONG varchar2(10),
  MLUONG number,
  CONSTRAINT PK_NV PRIMARY KEY (MANV),
  CONSTRAINT FK_NV_PB FOREIGN KEY (PHONG)
  REFERENCES PHONGBAN (MAPHG) )
```

24

24


 **Ngôn ngữ định nghĩa dữ liệu**
Tạo table, view (6)

❑ 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
nv.PHONG = 5
```

25

25

 **Ngôn ngữ định nghĩa dữ liệu**
Sửa cấu trúc table (7)

❑ 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_liệu_mới
```

– Ví dụ: sửa kiểu dữ liệu của cột Ghi chú

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

26

26

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

❑ 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
```

27

27

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

☐ Thêm ràng buộc toàn vẹn

ALTER TABLE <tên_bảng>
ADD CONSTRAINT
<tên_ràng_buộc>

- **UNIQUE** (tên_cột)
- **PRIMARY KEY** (tên_cột1,2,..)
- **FOREIGN KEY** (tên_cột)
REFERENCES tên_bảng
(cột_là_khóa_chính)
- **CHECK** (tên_cột
điều_kiện)

28

28

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)

29

29

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

30

30

 **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 table nhân viên
DROP TABLE NHANVIEN

- Ví dụ: xóa view TrucThuoc
DROP VIEW TrucThuoc

31

31


 **Ngôn ngữ thao tác dữ liệu**
Thêm dữ liệu vào bảng (1)

❑ Cú pháp
INSERT INTO tên_bảng VALUES (giá_trị_1,
giá_trị_2,..., giá_trị_n)
INSERT INTO tên_bảng (cột1, cột2) VALUES
(giá_trị_1, giá_trị_2)

- Ví dụ
INSERT INTO NHANVIEN VALUES ('001',
'Vuong', 'Ngoc', 'Quyen', 'Nu',
To_Date('22/10/1957','dd/mm/yyyy'), '450 Trung
Vuong, Ha Hoi', null, 'QL', 3000000)

32

32


 **Ngôn ngữ thao tác dữ liệu**
Sửa dữ liệu của bảng (2)

❑ Cú pháp
UPDATE tên_bảng SET cột_1 = giá_trị_1, cột_2 =
giá_trị_2 [WHERE điều_kiện]

- Ví dụ: Sửa họ nhân viên có mã số '001' thành
'Nguyen'
UPDATE NHANVIEN
SET HONV = 'Nguyen'
WHERE MANV='001'

33

33

 **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'
```

34

34

 **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
```

35

35


 **Ngôn ngữ thao tác dữ liệu**
Truy vấn dữ liệu – lệnh SELECT (5)

❑ Câu truy vấn tổng quát

```
SELECT [DISTINCT] tên_cột | hàm
FROM bảng
[WHERE điều_kiện]
[GROUP BY cột]
[HAVING điều_kiện]
[ORDER BY cột ASC | DESC]
```

36

36

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

☐ Toán tử so sánh

=

>

<

>=

<=


<>

☐ Toán tử logic: AND, OR, NOT

☐ Phép toán: +, -, *, /

37

37


 **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

38

38

 **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.


– SELECT * FROM NHANVIEN WHERE
To_number(To_char(NGSINH,'yyyy')) BETWEEN
1965 AND 1977

Ví dụ: Tìm nhân viên có lương không nằm trong khoảng
100000 đến 300000

– SELECT MANV, TENNV, TENPHG
FROM NHANVIEN n, PHONGBAN p
WHERE n.PHONG=p.MAPHG
AND MLUONG NOT BETWEEN 100000 AND
300000

39

39

 **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


```
SELECT * FROM NHANVIEN
WHERE MA_NQL IS NOT NULL
```

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

```
SELECT * FROM NHANVIEN
WHERE MA_NQL IS NULL
```

40

40

 **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 dụ

```
SELECT * FROM NHANVIEN
WHERE HONV LIKE 'Nguyen%'
```

41

41

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

☐ Toán tử IN

- Ví dụ

- SELECT * FROM NHANVIEN
WHERE PHONG IN ('NC','QL','DH')
- SELECT MANV, TENNV, DCHI
FROM NHANVIEN
WHERE MANV NOT IN (SELECT MANV
FROM PHANCONG)

42

42

Ngôn ngữ thao tác 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))
```

43

43

Ngôn ngữ thao tác dữ liệu
Mệnh đề GROUP BY (13)

□ 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).

44

44

Ngôn ngữ thao tác dữ liệu
Mệnh đề GROUP BY (14)

Quan hệ NV

Q	S
a	10
a	2
b	9
b	5
c	10
c	8
c	6
c	4
c	10
d	16
d	18
d	50

Chia các dòng thành các nhóm dựa trên tập thuộc tính chia nhóm

Q	Count(S)
a	2
b	2
c	5
d	3


Tương tự cho các hàm SUM, MIN, MAX, AVG

Các thuộc tính GROUP BY: Q

Câu SQL:
 Select Q, count(S)
 From NV
 Group by Q

45

45


 **Ngôn ngữ thao tác 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

46

46

 **Ngôn ngữ thao tác 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(MLUONG), MAX(MLUONG),
MIN(MLUONG), AVG(MLUONG) 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(MLUONG), MAX(MLUONG),
MIN(MLUONG), AVG(MLUONG)
FROM NHANVIEN, PHONGBAN
WHERE MAPHG=PHONG AND
TENPHG='Nghien cuu';
```

❑ Ví dụ: cho biết số lượng nhân viên

```
SELECT COUNT(*) FROM NHANVIEN;
```

47

47


 **Ngôn ngữ thao tác dữ liệu**
Mệnh đề GROUP BY (17)

❑ Ví dụ

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

48

48


 **Ngôn ngữ thao tác 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.

49

49


 **Ngôn ngữ thao tác 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 TENPHG, COUNT(*) so_luong_nv,
AVG(MLUONG) luong_tb
FROM NHANVIEN n, PHONGBAN p
WHERE n.PHONG = p.MAPHG
GROUP BY TENPHG
HAVING AVG(MLUONG) > 2000000
```

50

50

 **Ngôn ngữ thao tác 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, TENNV, SUM (THOIGIAN)
FROM NHANVIEN n, PHANCONG p
WHERE n.MANV = p.MANV
GROUP BY n.MANV, TENNV
HAVING SUM (THOIGIAN) >= ALL
      (SELECT SUM(THOIGIAN)
FROM PHANCONG
GROUP BY MANV)
```

51

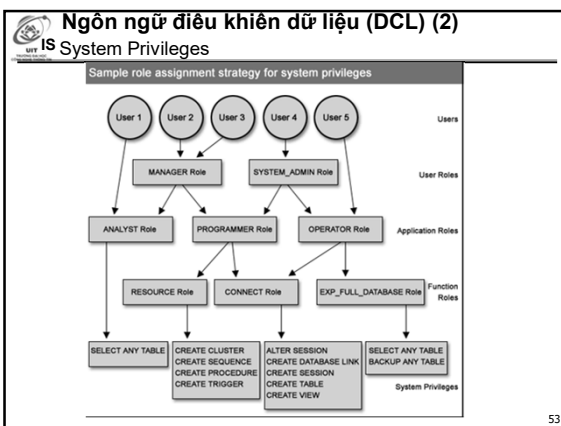
51

Ngôn ngữ điều khiển dữ liệu (DCL)
IS 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: dùng cấp phát quyền trên các đối tượng **tables, view, procedure** đến các user hoặc role.
 - REVOKE: dùng để thu hồi quyền đã cấp cho các user và role.
- ❑ 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,.... Quyền System privileges bao gồm 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, References, All trên các object như: tables, views, synonyms, procedure. Những quyền này được cấp bởi người sở hữu object.

52

52



53


Ngôn ngữ điều khiển dữ liệu (DCL) (3)
IS 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.

54


54

 **Ngôn ngữ điều khiển dữ liệu (DCL) (4)**

- ❑ Một User muốn tạo được schema riêng cho mình cần phải có 2 nhóm quyền hệ thống thích hợp là CONNECT đến cơ sở dữ liệu và quyền 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,....

55


55

 **Ngôn ngữ điều khiển dữ liệu (DCL) (5)**

- ❑ Note: The resource role has the UNLIMITED TABLESPACE priv, meaning it can use as much of any tablespace as it wants.
- ❑ 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”

56

56


 **Ngôn ngữ điều khiển dữ liệu (DCL) (6)**

Tạo người dùng

- ❑ **Cú pháp**
CREATE USER <user-name> NOT IDENTIFIED | IDENTIFIED BY <password> 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 Phung với password là P987654
 CREATE USER Phung IDENTIFIED BY P987654
 CREATE USER Phung1 IDENTIFIED BY P987654 default tablespace TEMP
 CREATE USER Phung1 IDENTIFIED BY P987654 default tablespace USERS

57

57


 **Ngôn ngữ điều khiển dữ liệu (DCL) (7)**
IS 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 Phung là p123
 ALTER USER Phung IDENTIFIED BY p123

❑ **Xóa người dùng**
Cú pháp: DROP USER <User-name> CASCADE
 Ví dụ: xóa người dùng Phung với tất cả object của user Phung, nếu Phung không có object thì không cần từ khóa CASCADE
 DROP USER Phung

58

58

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


❑ **Cú pháp**
CREATE ROLE <role-name> NOT IDENTIFIED | IDENTIFIED BY <password>
 Trong đó:
 - NOT IDENTIFIED: không đặt password cho role.
 - IDENTIFIED BY password: user phải nhập password để làm cho role có hiệu lực.
Lưu ý: 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

59

59


 **Ngôn ngữ điều khiển dữ liệu (DCL) (9)**
IS 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

60

60


 **Ngôn ngữ điều khiển dữ liệu (DCL) (10)**
IS 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]
Ví dụ: Grant **create synonym, create any table** to Phung
Grant **CONNECT, RESOURCE** to Phung
Grant **create synonym** to TaoSynonym

Cú pháp cấp phát quyền Object Privileges cho người dùng
GRANT privi1, privi2,.../ALL ON <TênObject> TO User1, User2,... [WITH GRANT OPTION]
Ví dụ: Grant **select, insert** on Scott.emp TO Phung

61

61

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

❑ **Cú pháp cấp phát quyền System Privileges cho Role**
GRANT tên_role_system_1, tên_role_system_2,... TO Role1, Role2,... [WITH GRANT OPTION]
Ví dụ: Grant **create synonym, create any table** to QuanTriSV
Grant **CONNECT, RESOURCE** to QuanTriSV


❑ **Cú pháp cấp phát quyền Object Privileges cho Role**
GRANT privi1, privi2,... [ON <TênObject>] TO Role1, Role2,... [WITH GRANT OPTION]

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

privi1, privi1,... 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,...

62

62

 **Ngôn ngữ điều khiển dữ liệu (DCL) (12)**
IS Cấp phát quyền


❑ **Cấp quyền truy xuất đến tất cả User, Role bằng lệnh**
GRANT privi1, privi2,.../ALL ON <TênObject> TO PUBLIC
Ví dụ: grant select on TênUser.TênTable to public

❑ **Ví dụ 1: cấp phát quyền cho user Phung**
GRANT INSERT, UPDATE ON TênUserTạoTableNhanVien.NHANVIEN TO Phung

❑ **Ví dụ 2: cấp phát quyền cho role QuanTriSV**
GRANT ALL ON TênUserTạoTableNhanVien.NHANVIEN TO QuanTriSV WITH GRANT ADMIN

63

63

 **Ngôn ngữ điều khiển dữ liệu (DCL) (13)**
IS Thu hồi quyền

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


Vi dụ: revoke select on TênUser.TênTable from public

☐ Ví dụ 1: thu hồi quyền từ user Phung
REVOKE INSERT, UPDATE ON
TênUserTạoTableNhanVien.NHANVIEN FROM Phung

☐ Ví dụ 2: thu hồi quyền của role QuanTriSV
REVOKE ALL ON TênUserTạoTableNhanVien.NHANVIEN
FROM QuanTriSV

64

64


 **Ngôn ngữ điều khiển dữ liệu (DCL) (14)**
IS Ví dụ đoạn PL/SQL cấp quyền

```
-- 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;
```

65

65

 **Ngôn ngữ điều khiển dữ liệu (DCL) (15)**
IS Truy vấn xem thông tin quyền

☐ 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;

66

66


 **Các lệnh giao tác (1)**
IS Transaction

GIAO TÁC: một Transaction là một giao tác trên CSDL bao gồm chuỗi các thay đổi (hành động) trên một hay nhiều table. Điều khiển Transaction bằng các lệnh sau.

- ❑ **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)

67


67

 **Các lệnh giao tác (2)**
IS Transaction

- ❑ **Ví dụ: minh họa cách dùng các lệnh SavePoint, RollBack, Commit**
begin
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;
insert into student values(4,'Nguyen Van D');
savepoint A; /* savepoint A trước đó ko để hiểu nữa */
insert into student values(5,'Nguyen Van E');
savepoint D;
end;
- ❑ select * from student; -> kết quả ? sinh viên
- ❑ rollback to savepoint A; //thời hồi lệnh ngược đến savepoint A
- ❑ select * from student; -> kết quả ? sinh viên

68

68

 **Các lệnh giao tác (3)**
IS Transaction


**1) Giá trị nào được thêm vào bảng NUMBERS?
Test trường hợp khác sử dụng lệnh drop table,
lệnh 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;
```

69

69

 **Các lệnh giao tác (4)**
IS 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;

```

70

 **Các lệnh giao tác (5)**
IS Transaction

LƯU 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 dụ:


```

CREATE TABLE student (id NUMBER, name
VARCHAR2(100));
INSERT INTO student VALUES (1, 'Nguyễn Văn A');
CREATE TABLE course (id NUMBER, name
VARCHAR2(100));
INSERT INTO course VALUES (1, 'Toán');
INSERT INTO course VALUES (2, 'Văn');
ROLLBACK;

```

Kết quả thực thi block trên?

71


 **Các đối tượng khác**
IS 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.

72


 **Các đối tượng khác**
IS 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 dụ:
CREATE SYNONYM nv **FOR** phuong.NHANVIEN;

❑ **Xóa Synonym:**
Cú pháp:
DROP SYNONYM tên-synonym;
Ví dụ:
DROP SYNONYM nv;

73

73

 **Các đối tượng khác**
IS 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.

❑ **Tạo Index**
Cú pháp: **CREATE [UNIQUE] INDEX** index_name **ON** table (column1 [, <column2>], ...)
Ví dụ:
CREATE UNIQUE INDEX i_cmnd **ON** NHAN_KHAU(SO_CMND);

74

74

 **Các đối tượng khác**
IS 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).

75

75

HUST
IS Index (3)

Các đối tượng khác

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

VENDOR_PARTS		
VEND ID	PART NO	UNIT COST
10-12	10-440	35
10-12	10-441	4.28
10-12	10-442	5.17
10-12	10-443	4.33
10-12	10-444	1.39
10-12	10-445	9.28
10-12	10-446	1.39
10-12	10-447	9.28

```
CREATE INDEX VP_INDEX
ON
VENDOR_PARTS(VEND_ID,
PART_NO);
```

❑ **Xóa index**

Cú pháp: DROP INDEX index_name;

Ví dụ: DROP INDEX i_cmdnd;

76


- **Các đối tượng khác**
- **IS Sequence (1)**
- **Sequence (giá trị được tạo tự động)**

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]
```

77



Các đổi tượng học khác

IS Sequence (2)

❑ Ví dụ


```
CREATE SEQUENCE sboheadon
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 (lệnh có tác dụng ở client, không thấy được giá trị tăng do user khác gửi)
- Giá trị kế tiếp của sequence: tên_sequence.NEXTVAL (lệnh có tác dụng trên server)

78

 **Các đối tượng khác**
IS 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 seq3 NOCACHE;
```

79

79

 **Các đối tượng khác**
IS Sequence (4)

Ví dụ

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

```
SELECT sohoaddon.currval from DUAL;
```

Truy cập giá trị tiếp theo của sequence sohoaddon bằng lệnh:


```
SELECT sohoaddon.nextval from DUAL; (DUAL là bảng tạm trong Oracle)
```

Sử dụng sequence trong một câu insert:

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

80

80

 **Các đối tượng khác**
IS 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)...

81

81


 **Các đối tượng khác**
IS 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 dụ: DROP SEQUENCE sohoaddon

82

82


 **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 tạo
❑ USER_TYPES : xem các type mình đã tạo
❑ USER_SEQUENCES : xem các sequence mình tạo
❑ USER_PROCEDURES : xem các procedure,function mình đã tạo
❑ USER_TRIGGERS : xem các trigger mình đã tạo
❑ USER_SYNONYMS : xem các synonyms mình đã tạo
❑ user_sys_privs: xem quyền của user hiện hành
❑ Ví dụ: select * from user_tables

83

83

 **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

84

84

 **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 dụ: 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, định-dạng-ngày)

Hiện 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

Hiện 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 (nếu viết tắt chữ “Mon” sẽ
 hiện thị tên tháng được viết tắt, ví dụ Dec, Nov, Jun,...)

85

85

 **MỘT SỐ HÀM THƯỜNG DÙNG TRONG ORACLE**

☐ SINH VIÊN XEM FILE WORD ĐI KÈM

86

86
