




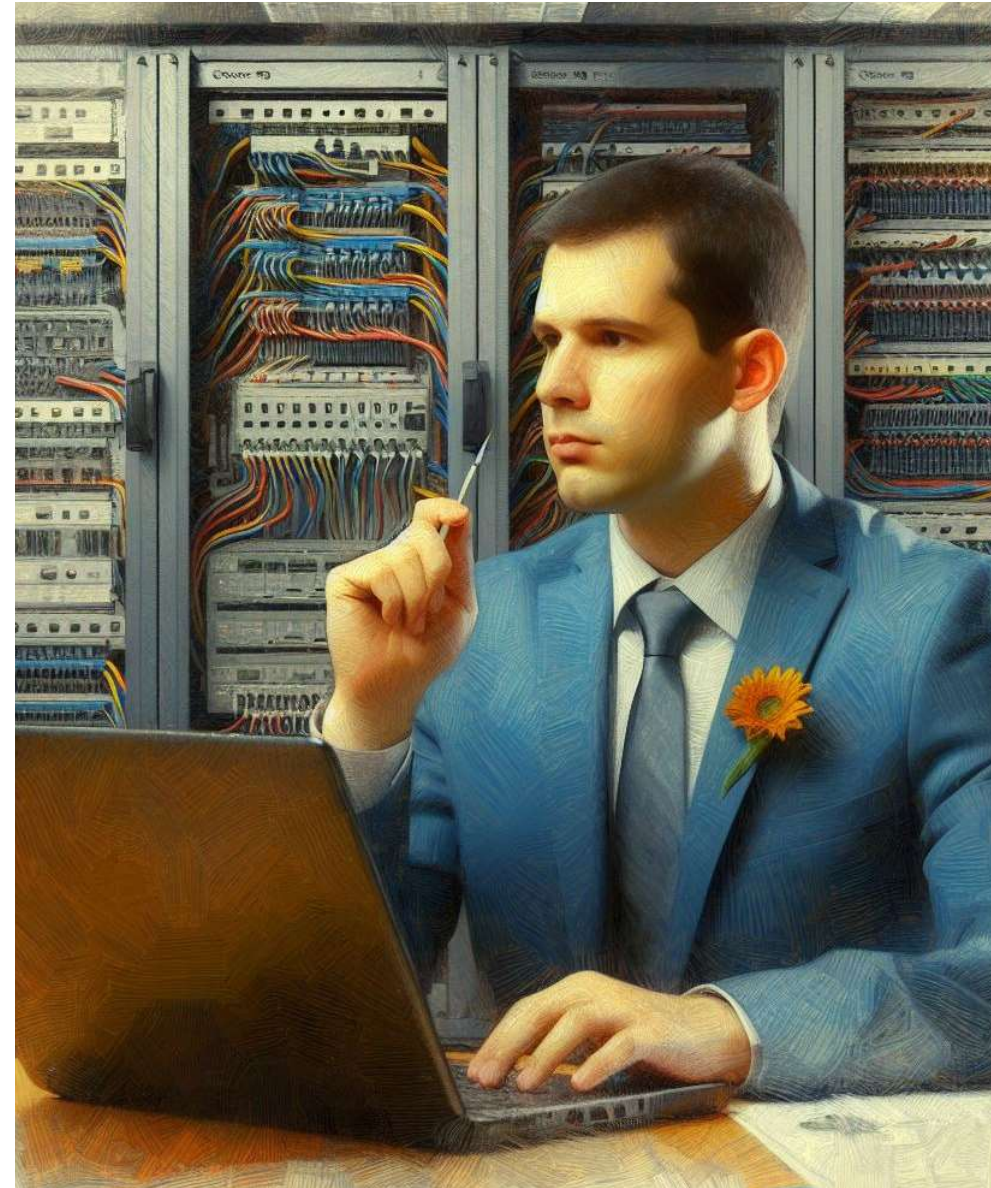
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

NGUYEN MINH NHUT
INFORMATION SYSTEM ENGINEERING

1

DBMS
noun [C]
UK  /,di:.bi:.em'es/ US  /,di:.bi:.em'es/ [Add to word list](#) 

abbreviation for database management system: a set of computer programs for allowing large amounts of information to be put into a computer and for organizing it so that it can be searched, examined, or printed easily and quickly



Database Management and Administration



2

CHƯƠNG 2

NGÔN NGỮ PL SQL

Oracle's Large Business Database



Trong chương này chúng ta sẽ học những nội dung:

- Store Procedure là gì? Function là gì?
- Biến Kiểu dữ liệu trong ngôn ngữ PL/SQL
- Viết Trigger/Function đơn giản
- Xử lý đơn dòng
- Xử lý nhiều dòng



PHẦN 1

STORE PROCEDURE VÀ FUNCTION

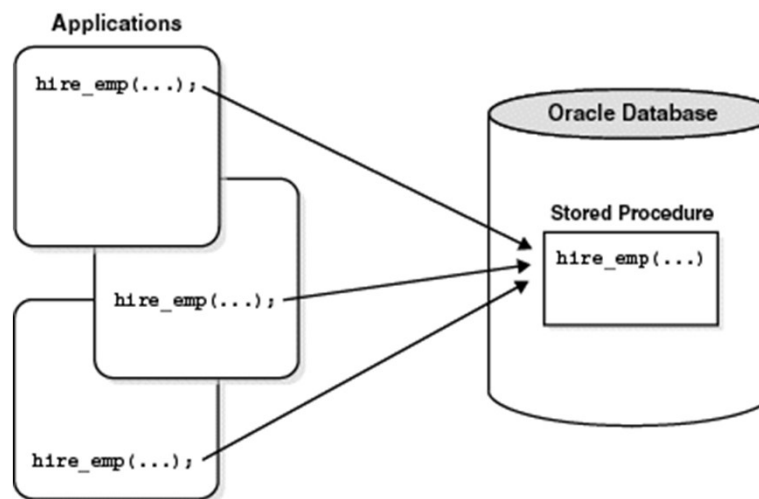
Store Procedure là gì? Tại sao phải dùng Store? Function là gì? Tại sao sử dụng Function? Định nghĩa một Simple Database đơn giản. Giải thích một số Store và Procedure đơn giản.



Phần 1. Store Procedure và Function

• Giới thiệu về Store Procedure

- **Store Procedure (Thủ tục)**: là một khối mã SQL đã được đặt tên và lưu trữ trong cơ sở dữ liệu để thực hiện một loạt các hành động hoặc tác vụ cụ thể. Nó chứa một tập hợp các câu lệnh SQL và có thể bao gồm các cấu trúc điều khiển để thực hiện các chức năng phức tạp.



Phần 1. Store Procedure và Function



• Giới thiệu về Store Procedure

- Stored Procedure có thể **chứa tham số đầu vào và đầu ra**, cung cấp **tính tái sử dụng cao** và **quản lý mã nguồn** một cách hiệu quả trong môi trường cơ sở dữ liệu.
- **Đặc điểm của Store Procedure**
 - Có thể kiểu trả về hoặc **không cần thiết**. (OUTPUT là tùy ý)
 - Store Procedure được **thực hiện nhiều hành động**. (**INSERT, DELETE, UPDATE**)
 - Có thể chứa các câu lệnh TRANSACTION như **COMMIT** và **ROLLBACK**
- **Ví dụ:** Thủ tục đăng ký khách hàng mới **proc_register_customer**
INSERT khách hàng mới (Customer) → INSERT Lưu lại lịch sử đăng ký khách hàng (LogSystem) → UPDATE số lượng khách hàng trong hệ thống (System)

Phần 1. Store Procedure và Function



- **Đặc điểm của Store Procedure**

- Có thể gọi từ một **Store Procedure** khác hay là một **Sub Store**
- Không thể sử dụng các mệnh đề **SELECT/WHERE/HAVING** một cách trực tiếp mà phải thông qua **INSERT DELETE hoặc UPDATE** với trường hợp không có giá trị **OUTPUT**.
- Có thể sử lý **Try Catch** và **EXCEPTION** trong Store Procedure

Phần 1. Store Procedure và Function



• Giới thiệu về Function

- **Function (Hàm):** là một khối mã SQL đã được đặt tên và lưu trữ trong cơ sở dữ liệu để thực hiện tính toán một đoạn mã và kết quả. Nhưng không cần nhất thiết phải câu query phức tạp.
- **Ví dụ 2:** Tính mức lương trung bình trung bình của nhân viên với mã được nhập đầu vào
 - INPUT: customer_id
 - OUTPUT: avg_salary
- **Đặc điểm của Function (Hàm)**
 - Phải có giá trị **TRẢ VỀ** (OUTPUT) đơn hoặc bảng
 - **Không** được phép sử dụng TRANSACTION trong Function (Vì nó thiên về SELECT)
 - **Không** thể gọi Store Procedure, Function con bên trong

Phần 1. Store Procedure và Function



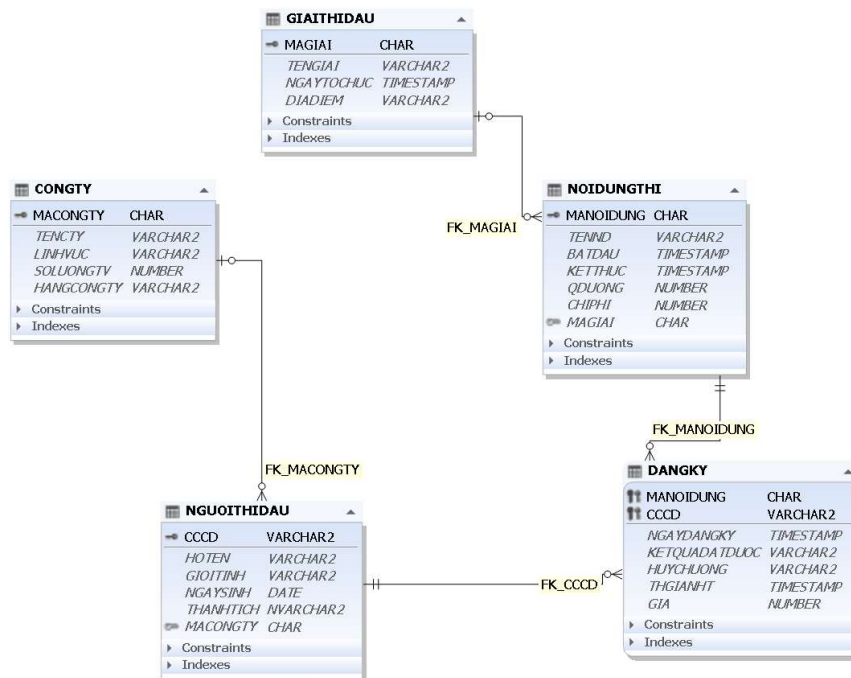
- Giới thiệu về Function

- Đặc điểm của Function (Hàm)

- Phải có giá trị **TRẢ VỀ** (OUTPUT) đơn hoặc bảng
 - **Không** được phép sử dụng TRANSACTION trong Function (Vì nó thiên về SELECT)
 - **Không** thể gọi Store Procedure, Function con bên trong
 - Không xử lý **Try-Catch** và **EXCEPTION**
 - Có thể dùng các câu lệnh **SELECT/WHERE/HAVING**

Phần 1. Store Procedure và Function

• Ví dụ một Store Procedure đơn giản



Thủ tục: Đăng ký tham gia thi đấu

Input: \$CCCD, \$MANOIDUNG

```
1 Calculate $Gia From NOIDUNGTHI.CHIPHI*Rate
2 INSERT DANGKY (
3     MANOIDUNG = $MANOIDUNG
4     CCCD = $CCCD
5     NGAYDANGKY = SYSDATE
6     KETQUATHIDATDUOC = NULL
7     HUYCHUONG = NULL
8     THGIANHT = NULL
9     GIA = $GIA)
10 IF SumPrice(MACONGTY of $CCCD) > 1 billions
11     UPDATE HANGCONGTY = 'Gold'
12 ELSEIF (SumPrice(MACONGTY of $CCCD) > 5
13         billions
14     UPDATE HANGCONGTY = 'Diamond'
```

Phần 1. Store Procedure và Function

- Ví dụ một Function đơn giản

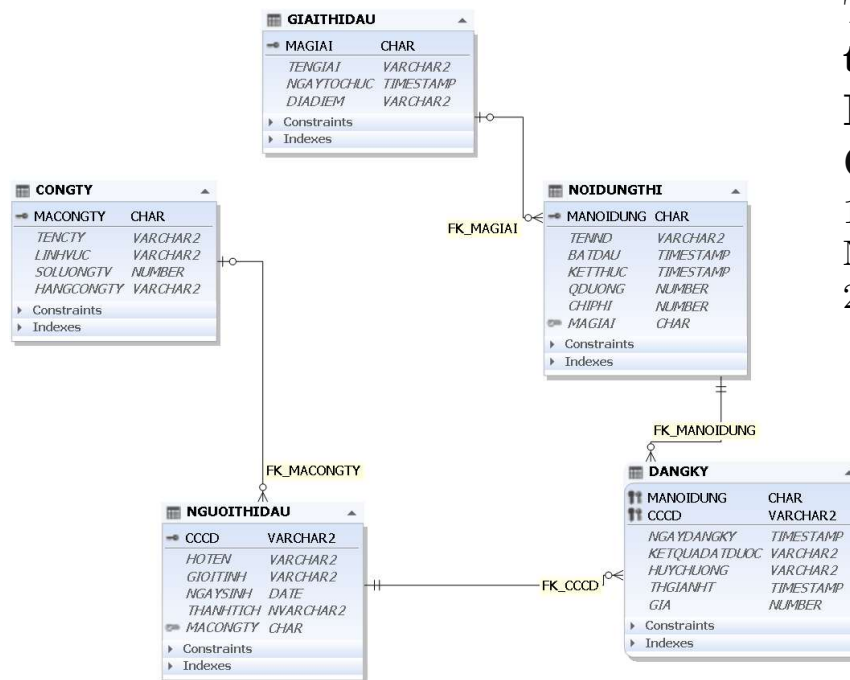
Thủ tục: Tính tổng chi phí mà công ty đã trả dựa vào mã MACONGTY

Input: \$MACONGTY

Output: \$SumPrice

1 \$SumPrice = SUM(GIA) FROM JOIN (CONGTY, NGUOITHIDAU, DANGKY);

2 RETURN \$SumPrice;





PHẦN 2

NGÔN NGỮ PL/SQL, BIẾN VÀ KIỂU DỮ LIỆU

Ngôn ngữ PL/SQL trong Oracle, giới thiệu về biến, kiểu dữ liệu trong Oracle.



Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Giới thiệu về ngôn ngữ PL/SQL

- Ngôn ngữ PL/SQL viết tắt **Procedural Language/Structured Query Language** là ngôn ngữ lập trình hướng thủ tục được sử dụng trong Hệ quản trị CSDL Oracle.
- Một đoạn block tham khảo PL/SQL

```
01 -- This is an anonymous procedure, so it has no name
02 DECLARE
03     /*
04     You declare local cursors, variables, and methods here,
05     but you don't need to have a declaration section.
06     */
07 BEGIN
08     -- You write your executable code here
09
10     NULL; -- Ahhh, you've got to have at least one command!
11 EXCEPTION
12     when NO_DATA_FOUND then
13         raise_application_error(-20000,
14             'Hey, This is in the exception-handling section!');
15 END;
16 /
17 -- the forward slash on a line by itself says execute this procedure
```


Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Giới thiệu về ngôn ngữ PL/SQL

- **DECLARE:** Phần khai báo có thể là biến con trỏ, biến hoặc các phương thức
- **BEGIN ... END:** Nơi viết các lệnh PL/SQL
- **EXCEPTION:** Nơi xử lý ngoại lệ

```
01 -- This is an anonymous procedure, so it has no name
02 DECLARE
03     /*
04     You declare local cursors, variables, and methods here,
05     but you don't need to have a declaration section.
06     */
07 BEGIN
08     -- You write your executable code here
09
10     NULL; -- Ahhh, you've got to have at least one command!
11 EXCEPTION
12     when NO_DATA_FOUND then
13         raise_application_error(-20000,
14             'Hey, This is in the exception-handling section!');
15 END;
16 /
17 -- the forward slash on a line by itself says execute this procedure
```

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Khai báo biến

- Cách đặt tên biến

Prefix	Data Type
c_	CURSOR
d_	DATE
n_	NUMBER
r_	ROW
t_	TABLE
v_	VARCHAR2

```
declare

    n_id          number;
    v_name        varchar2(100);
    d_birth_date  date;
    v_gender      varchar2(30);

begin
    ...
end;
```

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Khai báo biến

- Giả sử muốn khai báo biến cùng kiểu dữ liệu với **một thuộc tính trong bảng**
- Ví dụ
 - CUSTOMER(CusID, CusName, Birthday, Address)
- Muốn khai báo biến **n_customer_name** cùng kiểu dữ liệu với **CusName**

DECLARE

n_customer_name CUSTOMER.CusName%TYPE;

- Mục tiêu của việc khai báo dựa vào thuộc tính bảng là đảm bảo cho việc khai báo đồng nhất **với kiểu dữ liệu trong bảng**, tránh trường hợp mất tính nhất quán dữ liệu.

```
01 declare
02   n_id          WORKERS.id%TYPE          := 1;
03   n_worker_type_id WORKERS.worker_type_id%TYPE := 3;
04   v_external_id  WORKERS.external_id%TYPE  := '6305551212';
05   v_first_name   WORKERS.first_name%TYPE   := 'JANE';
06   v_middle_name  WORKERS.middle_name%TYPE  := 'E';
07   v_last_name    WORKERS.last_name%TYPE    := 'DOE';
08   v_name         WORKERS.name%TYPE         := 'JANEDOE';
09   d_birth_date   WORKERS.birth_date%TYPE   :=
10     to_date('19800101', 'YYYYMMDD');
```

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

- Kiểu dữ liệu và các nhóm kiểu dữ liệu trong ngôn ngữ PL/SQL

Nhóm Datatype	Mô tả và đại diện
Numeric	Giá trị là số, có khả năng tính toán được các phép toán Ví dụ: Float, Long, Number(p, s),...
Character	Giá trị thường là dạng ký tự, chuỗi Ví dụ: Varchar2(n), Nvarchar2(n),...
Datetime	Thường lưu giá trị kiểu ngày tháng. Ví dụ: Date,...
Timestamp	Thường lưu giá trị thời gian có giờ phút giây Ví dụ: Timestamps,...
Cursor	Kiểu dữ liệu con trỏ trong Oracle

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Các toán tử trong PL/SQL (Operator)

Nhóm Toán tử	Mô tả và đại diện
Tính toán	Thực hiện các phép tính toán trên CSDL Ví dụ: +, -, *, /
Phép cộng chuỗi	Cộng hai chuỗi bất kì bằng phép toán Ví dụ: "A" "B" → Kết quả: AB
COLLATE	Thường được đặt ở mệnh đề ORDER, mục tiêu so sánh theo bộ mã mong muốn. Mặc định của Oracle là Alphabet nên muốn so sánh theo bộ mã khác dùng COLLATE ORDER <Column name> COLLATE <Bảng mã>
Phép toán tập hợp	Thường được dùng để tính các phép toán tập hợp: Ví dụ: UNION, UNION ALL, MINUS, INTERSECT

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Cấu trúc điều khiển – Cấu trúc rẽ nhánh

– Cấu trúc rẽ nhánh trong PL/SQL Oracle cũng giống như trong ngôn ngữ lập trình khác

```
IF <Điều kiện> THEN
    //Thực thi logics bên trong
END IF;
```

– Cấu trúc rẽ nhánh thêm điều kiện ELSE IF, ELSE và CASE ... WHEN

```
IF <Điều kiện 1> THEN
    //Thực thi logic 1 bên trong
ELSEIF <Điều kiện 2> THEN
    //Thực thi logic 2 bên trong
ELSE
    //Thực thi logic 3 bên trong
END IF;
```

```
CASE <giá trị>
    WHEN <điều kiện 1> THEN //logic1
    WHEN <điều kiện 2> THEN //logic2
    ...
    ELSE //logic_final
END
```

Phần 2. Ngôn Ngữ PL/SQL, biến và kiểu dữ liệu

• Cấu trúc điều khiển – Cấu trúc lặp

- Lặp với số lần đã biết trước FOR ... LOOP

```
FOR index IN <array | CURSOR>  
LOOP  
    //Thực thi logic  
END LOOP;
```

- Lặp với số lần chưa biết trước WHILE ... LOOP

```
WHILE <điều kiện>  
LOOP  
    //Thực thi logic  
END LOOP;
```



PHẦN 3

TRIGGER VÀ CON TRỎ TRONG ORACLE

20

Trigger và con trỏ trong Oracle có nhiều đặc điểm của PL/SQL, thông qua đó thực hành trên bài tập quản lý chung cư để nắm rõ bản chất Trigger và con trỏ trong Oracle.



Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

- Cách thức khai báo con trỏ trong Oracle

- Khai báo con trỏ trong Oracle

```
DECLARE  
    CURSOR <cursor_name> IS <Query>
```

- Đóng mở CURSOR, Lặp con trỏ và điều kiện dừng

```
OPEN <cursor_name>;  
FETCH <cursor_name> INTO <cursor_var>;  
WHILE <cursor_name>%FOUND LOOP  
    //Thực thi bên trong con trỏ  
    FETCH <cursor_name> INTO <cursor_var>;  
END LOOP;  
CLOSE <cursor_name>;
```

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

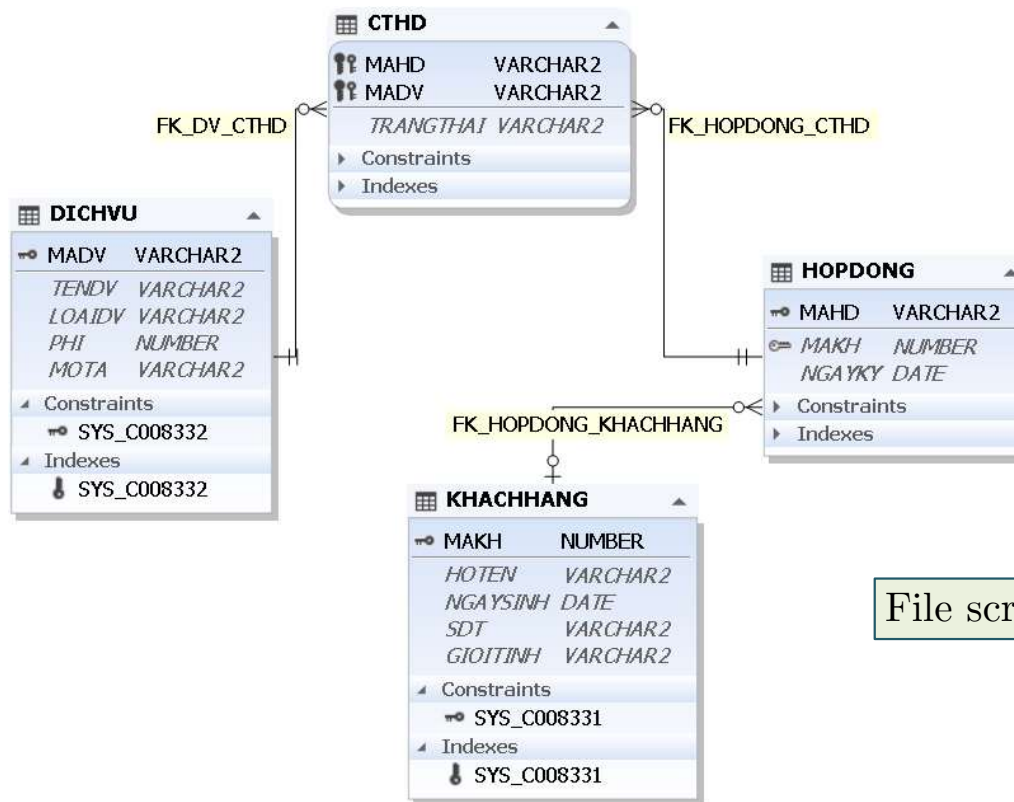
- Cấu trúc một Trigger trong Oracle

Cấu trúc một Trigger

```
CREATE [OR REPLACE TRIGGER] <trigger_name>
BEFORE INSERT [OR UPDATE OR DELETE] ON <table_name>
FOR EACH ROW
DECLARE
    // Khai báo biến cho Trigger
    -- DBMS_OUTPUT.PUT_LINE('ERROR: LOI NGAY DANG KI!');
    -- Nếu muốn hủy thêm hoặc sửa, hãy sử dụng RAISE_APPLICATION_ERROR
    -- RAISE_APPLICATION_ERROR(-20001, 'ERROR: LOI NGAY DANG KI!');
BEGIN
    // Nội dung bên trong Trigger
END;
/
```

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

- Giới thiệu về lược đồ CSDL Quản lý Dịch vụ Chung Cư



File script: [Lecture2_Trigger.sql](#)

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

- **Ràng buộc toàn vẹn**

- Ngày đăng ký hợp đồng phải lớn hơn ngày sinh của khách hàng.
- **Bối cảnh:** HOPDONG, KHACH HANG
- **Nội dung:**

$$\exists kh \in KHACHHANG, \forall hd \in HOPDONG : kh.MAKH = hd.MAKH \\ \wedge kh.NGAYSINH < hd.NGAYDK$$

- **Bảng tầm ảnh hưởng:**

	THÊM	XÓA	SỬA
KHACHHANG	-	-	+(NgaySinh)
HOPDONG	+	-	+(NgayDK)

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư


- Viết Trigger cho trường hợp thêm, sửa hợp đồng



```
CREATE OR REPLACE TRIGGER themsửa_hdong
BEFORE INSERT OR UPDATE ON HOPDONG
FOR EACH ROW
DECLARE
    v_NGAYSINH DATE;
BEGIN
    -- Lấy thông tin NGAYSINH từ bảng KHACHHANG
    SELECT NGAYSINH INTO v_NGAYSINH
    FROM KHACHHANG
    WHERE MAKH = :NEW.MAKH;
    -- Kiểm tra điều kiện và xử lý
    IF :NEW.NGAYKY < v_NGAYSINH THEN
        DBMS_OUTPUT.PUT_LINE('ERROR: LOI NGAY DANG KI!');
        -- Nếu muốn hủy thêm hoặc sửa, hãy sử dụng RAISE_APPLICATION_ERROR
        RAISE_APPLICATION_ERROR(-20001, 'ERROR: LOI NGAY DANG KI!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('THEM THANH CONG');
    END IF;
END;
/
```

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

- Viết Trigger cho trường hợp thêm hợp đồng



```
CREATE OR REPLACE TRIGGER themsua_hdong
BEFORE INSERT OR UPDATE ON HOPDONG
FOR EACH ROW
DECLARE
    v_NGAYSINH DATE;
BEGIN
    -- Lấy thông tin NGAYSINH từ bảng KHACHHANG
    SELECT NGAYSINH INTO v_NGAYSINH
    FROM KHACHHANG
    WHERE MAKH = :NEW.MAKH;

    -- Kiểm tra điều kiện và xử lý
    IF :NEW.NGAYKY < v_NGAYSINH THEN
        -- Nếu muốn hủy thêm hoặc sửa, hãy sử dụng RAISE_APPLICATION_ERROR
        RAISE_APPLICATION_ERROR(-20001, 'ERROR: LOI NGAY DANG KI!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('THEM THANH CONG');
    END IF;
END;
/

SELECT * FROM HOPDONG;
/** Truong hop that bai*/
INSERT INTO HOPDONG VALUES('HD06', 1, TO_DATE('26/4/1999', 'DD/MM/YYYY'));
/** Truong hop thanh cong*/
INSERT INTO HOPDONG VALUES('HD06', 1, TO_DATE('26/4/2024', 'DD/MM/YYYY'));
```

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư


- Viết Trigger cho trường hợp thêm hợp đồng

```
SELECT * FROM HOPDONG;  
/** Truong hop that bai*/  
INSERT INTO HOPDONG VALUES('HD06', 1, TO_DATE('26/4/1999', 'DD/MM/YYYY'));  
/** Truong hop thanh cong*/  
INSERT INTO HOPDONG VALUES('HD06', 1, TO_DATE('26/4/2024', 'DD/MM/YYYY'));
```

Error List	
1 Errors	0 Warnings
0 Messages	1 In Active Document
1 of 1 items displayed	
Description	
1	ORA-20001: ERROR: LOI NGAY DANG KI ORA-06512: at "QLTIENNET.THEMSUA_HDONG", line 12 ORA-04088: error during execution of trigger 'QLTIENNET.THEMSUA_HDONG'

Phần 3. Mô tả lược đồ CSDL Quản lý Chung Cư

- Viết Trigger cho trường hợp sửa khách hàng



```
DECLARE
CURSOR cursorDK IS
    SELECT MAHD FROM HOPDONG WHERE MAKH = :NEW.MAKH;
BEGIN
    OPEN cursorDK;
    FETCH cursorDK INTO v_ MAHDDK;
    WHILE cursorDK%FOUND LOOP
        -- Lấy thông tin NGAYKY từ bảng HOPDONG
        SELECT NGAYKY INTO v_ NGAYKY
        FROM HOPDONG
        WHERE MAHD = v_ MAHDDK;
        -- Kiểm tra điều kiện và xử lý
        IF v_ NGAYKY < :NEW.NGAYSINH THEN
            RAISE_APPLICATION_ERROR(-20001, 'ERROR: LOI NGAY DANG KI!');
            ROLLBACK;
        EXIT;
    END IF;
    FETCH cursorDK INTO v_ MAHDDK;
END LOOP;
CLOSE cursorDK;
```




PHẦN 4

FUNCTION VÀ PROCEDURE TRONG ORACLE

Procedure và Function là 2 keywords quan trọng trong PL/SQL Oracle với nhiều điểm mạnh, giúp chương trình hoạt động và giao tiếp nhanh hơn. Lập trình Function và Procedure thuộc lĩnh vực lập trình CSDL.

29



Phần 4. Procedure và Function trong Oracle

- Cấu trúc một Procedure

```
CREATE [OR REPLACE PROCEDURE] <procedure_name>(  
    v_bien1 IN <datatype>,  
    v_bien2 IN <datatype>  
    ....  
) AS BEGIN  
    // Nội dung bên trong Procedure  
END;
```

- Cấu trúc một Function

```
CREATE [OR REPLACE FUNCTION] <function_name>(  
    v_bien1 IN <datatype>,  
    v_bien2 IN <datatype>  
) RETURN <datatype> AS <v_return> <datatype>  
BEGIN  
    // Nội dung bên trong Function  
END;
```

Phần 4. Procedure và Function trong Oracle

- Viết Procedure – Thêm một khách hàng

```
/**Viết Procedure thêm một khách hàng*/
CREATE OR REPLACE PROCEDURE ThemKhachHang(
    p_MAKH IN NUMBER,
    p_HOTEN IN VARCHAR2,
    p_NGAYSINH IN DATE,
    p_SDT IN VARCHAR2,
    p_GIOITINH IN VARCHAR2
) AS
BEGIN
    INSERT INTO KHACHHANG VALUES (p_MAKH, p_HOTEN, p_NGAYSINH, p_SDT, p_GIOITINH);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Khach hang da duoc them thanh cong!');
END;
```

- Gọi Procedure

```
/**Gọi Procedure Them Khach Hang*/
-- Khởi PL/SQL để thực thi Procedure
DECLARE
    v_MAKH NUMBER := 4;
    v_HOTEN VARCHAR2(50) := 'Nguyen Van A';
    v_NGAYSINH DATE := TO_DATE('01/01/1990', 'DD/MM/YYYY');
    v_SDT VARCHAR2(11) := '0123456789';
    v_GIOITINH VARCHAR2(3) := 'Nam';
BEGIN
    ThemKhachHang(v_MAKH, v_HOTEN, v_NGAYSINH, v_SDT, v_GIOITINH);
END;
/

SELECT * FROM KHACHHANG;
```

MAKH NUMBER	HOTEN VARCHAR2(50)	NGAYSINH DATE	SDT VARCHAR2(11)	GIOITINH VARCHAR2(3)
1	Le Hoang Nguyen	03/25/ 2001 12:00:00 AM	0773577204	Nam
2	Phan Anh Nhat	03/26/ 2001 12:00:00 AM	0352977905	Nam
3	Tran Ngoc Giao	02/25/ 2001 12:00:00 AM	(null)	Nu
4	Nguyen Van A	01/01/ 1990 12:00:00 AM	0123456789	Nam

Phần 4. Procedure và Function trong Oracle

- Viết Procedure – Cập nhật ngày ký hợp đồng

```
/** Question 2 - Cập nhật ngày ký hợp đồng*/  
CREATE OR REPLACE PROCEDURE CapNhatNgayKyHopDong(  
    p_MAHĐ IN VARCHAR2,  
    p_NGAYKY IN DATE  
) AS  
BEGIN  
    UPDATE HOPDONG  
    SET NGAYKY = p_NGAYKY  
    WHERE MAHD = p_MAHĐ;  
    COMMIT;  
    DBMS_OUTPUT.PUT_LINE('Ngày ký hợp đồng đã được cập nhật thành công!');  
END;  
/
```

- Gọi Procedure

```
-- Khối PL/SQL để gọi Procedure  
DECLARE  
    v_MAHĐ VARCHAR2(10) := 'HD01'; -- Đặt giá trị phù hợp  
    v_NGAYKY DATE := CURRENT_DATE; -- Đặt giá trị phù hợp  
BEGIN  
    CapNhatNgayKyHopDong(v_MAHĐ, v_NGAYKY);  
END;  
/
```

MAHD VARCHAR2(10)	MAKH NUMBER	NGAYKY DATE
HD01	1	04/26/ 2020 12:00:00 AM
HD02	2	05/26/ 2020 12:00:00 AM
HD03	2	05/29/ 2020 12:00:00 AM
HD04	1	05/15/ 2020 12:00:00 AM
HD05	1	05/16/ 2020 12:00:00 AM
HD06	1	04/26/ 2024 12:00:00 AM

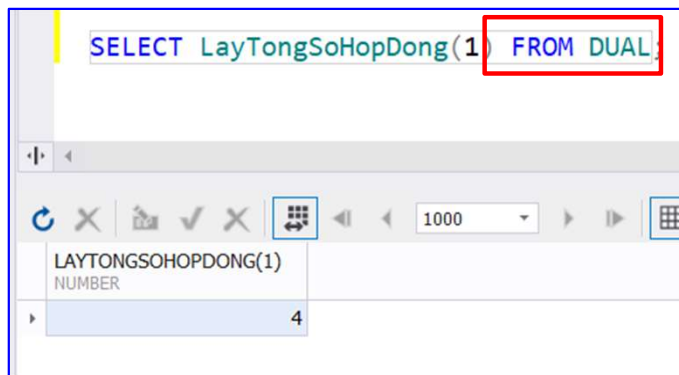
MAHD VARCHAR2(10)	MAKH NUMBER	NGAYKY DATE
HD01	1	12/28/ 2023 10:17:27 PM
HD02	2	05/26/ 2020 12:00:00 AM
HD03	2	05/29/ 2020 12:00:00 AM
HD04	1	05/15/ 2020 12:00:00 AM
HD05	1	05/16/ 2020 12:00:00 AM
HD06	1	04/26/ 2024 12:00:00 AM

Phần 4. Procedure và Function trong Oracle

- Viết Function - Lấy Tổng Số Hợp Đồng của Khách Hàng

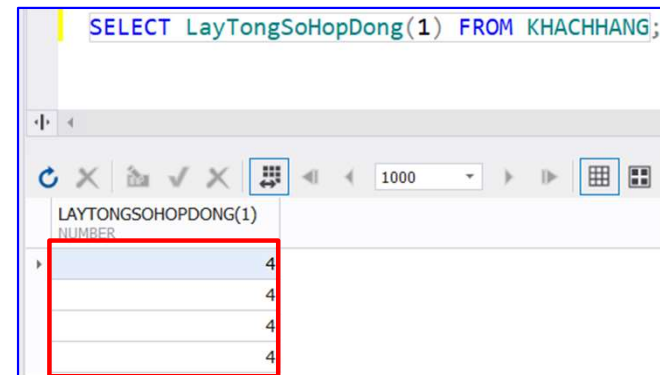
```
/** Question 3 - Viết Function lấy tổng hợp đồng khách hàng*/  
CREATE OR REPLACE FUNCTION LayTongSoHopDong(  
    p_MAKH IN NUMBER  
) RETURN NUMBER AS  
    v_TongSoHopDong NUMBER;  
BEGIN  
    SELECT COUNT(*) INTO v_TongSoHopDong  
    FROM HOPDONG  
    WHERE MAKH = p_MAKH;  
  
    RETURN v_TongSoHopDong;  
END;  
/
```

- Gọi Function



The screenshot shows a SQL query in the editor: `SELECT LayTongSoHopDong(1) FROM DUAL;`. The result grid below shows a single row with the value 4.

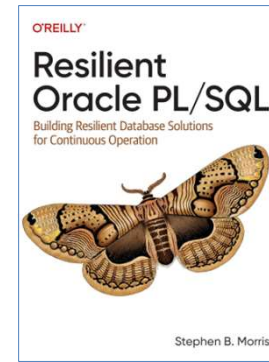
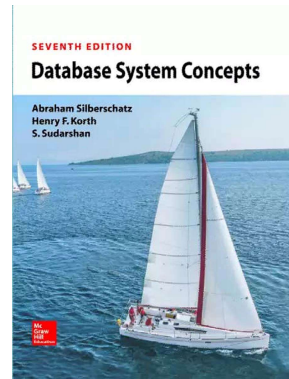
LAYTONGSOHOPDONG(1)
NUMBER
4



The screenshot shows a SQL query in the editor: `SELECT LayTongSoHopDong(1) FROM KHACHHANG;`. The result grid below shows four rows, each with the value 4.

LAYTONGSOHOPDONG(1)
NUMBER
4
4
4
4

Reference



- [1] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, 2020.
- [2] Stephen B. Maris, "Resilient Oracle PL/SQL 1st Edition", O'Reilly, 2023



CẢM ƠN ĐÃ THEO DÕI



ftisu.vn



minhnhut.ftisu@gmail.com



0939013911