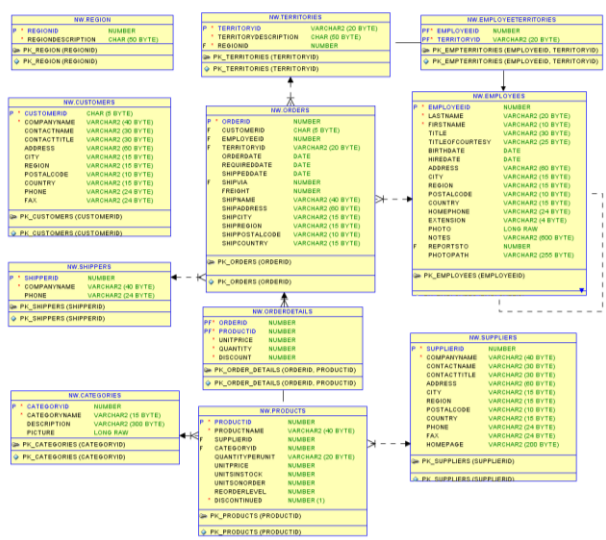
Laboratorio di PL/SQL

Prof. Alessandra Lumini
Alma Mater Studiorum - Università di Bologna

Per la sintassi PL/SQL:

- ORACLE 11g Rel. 2 – PL/SQL Language Reference

Schema DB



Procedure: definizione e call

- Una stored procedure è un blocco di codice PL/SQL dotato di un nome che viene mantenuto all'interno del database (procedure/funzioni)

```
CREATE OR REPLACE PROCEDURE nome_procedura
[(parametri)] IS
       Definizioni;
BEGIN
       Corpo procedura;
END;
```

```
CREATE FUNCTION nome_funzione [(parametri)] RETURN
tipo_dato IS
…
```

- Una procedura può essere richiamata utilizzando il comando call

```
CALL nome_procedura([parametri]);
```


La procedura «Hello world»

- Procedura per stampare a video la stringa «Hello world»

```
CREATE OR REPLACE PROCEDURE HELLOWORLD AS
v1 varchar2(12) :='Hello World!';
BEGIN
  DBMS_OUTPUT.PUT_LINE (v1);
END HELLOWORLD;
```

1. Scrivere una procedura che stampi in output la stringa LABORATORIO DI BASI DATI come concatenazione di 4 variabili

Query con una singola riga

- Procedura per stampare a video il numero di prodotti

```
CREATE OR REPLACE PROCEDURE NUMPRODOTTI AS
nProd NUMBER(5,0);
BEGIN
  SELECT count(*) into nProd FROM NW.PRODUCTS ;
  DBMS_OUTPUT.PUT_LINE ('Numero di Prodotti:' || nProd);
END;
```

2. Scrivere una procedura che stampi il numero di ordini e il ricavo totale

5

Uso di condizioni

- Procedura per stampare il numero di ordini N gestiti da un impiegato (in input). Se N>100 stampa «high», se minore di 50 stampa «low» altrimenti «medium»

```
create or replace PROCEDURE ProdImp(idImp number) AS
nOrd NUMBER(5,0);
BEGIN
  SELECT count(*) into nOrd FROM NW.ORDERS WHERE EmployeeID=idImp ;
  IF (nOrd) > 100 THEN
  DBMS_OUTPUT.PUT_LINE ('High:' || nOrd);
  ELSIF (nOrd) < 50 THEN
        DBMS_OUTPUT.PUT_LINE ('Low:' || nOrd);
  ELSE
        DBMS_OUTPUT.PUT_LINE ('Medium:' || nOrd);
  END IF;
END;
```

3. Scrivere una procedura che classifica un cliente in base al totale acquistato (best >100K /standard/worst <5K)

6

Gestione delle eccezioni

- Stampa dei dati di un cliente dato il nome. Gestire l'assenza del cliente o la presenza di più di un record

```
CREATE OR REPLACE PrintCliente(Nome VARCHAR2) AS
vCliente NW_CUSTOMERS%ROWTYPE;
BEGIN
  SELECT * into vCliente FROM NW_CUSTOMERS WHERE COMPANYNAME
LIKE Nome;
  DBMS_OUTPUT.PUT_LINE ('Cliente: ' || vCliente.COMPANYNAME
|| ' ID: ' || vCliente.CUSTOMERID );
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('Cliente non trovato');
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE ('Nome cliente non univoco');
END;
```

4. Scrivere una procedura che calcoli il totale degli ordini effettuati in una certa data. Gestire l'assenza di ordini.

Uso di cicli

- Stampa N date a partire da una data iniziale

```
CREATE OR REPLACE PROCEDURE PrintDates(iDate DATE, nPrint NUMBER) IS
    v_counter    NUMBER(2) := 0;
BEGIN
  LOOP
    EXIT WHEN v_counter >= nPrint;
 DBMS_OUTPUT.PUT_LINE(TO_CHAR(iDate+v_counter,'DD FMMonth YYYY'));
    v_counter := v_counter + 1;
  END LOOP;
END;

--oppure
for v_counter in 1 .. nPrint
LOOP
DBMS_OUTPUT.PUT_LINE(TO_CHAR(initDate+v_counter,'DD FMMonth YYYY'))
END LOOP;
```

5. Scrivere una procedura che stampa il totale di N ordini a partire da IDOrd (N e IDOrd in input). Gestire con una eccezione l'assenza di uno o più ordini

I cursori – ciclo FOR

- Definire un cursore per visualizzare le categorie

```
CREATE PROCEDURE PrintCategories IS
CURSOR cCat IS
SELECT CATEGORYID, CATEGORYNAME, DESCRIPTION FROM NW_CATEGORIES;

BEGIN
  FOR vCat IN cCat
  LOOP -- implicit open/fetch
    DBMS_OUTPUT.PUT_LINE(vCat.CATEGORYNAME || ': ' ||
        vCat.DESCRIPTION);
  END LOOP; -- Chiusura implicita
END;
```

6. Definire un cursore per stampare i corrieri e quanti ordini hanno gestito

9

I cursori – FETCH esplicito

- Definire un cursore per visualizzare le categorie

```
CREATE PROCEDURE PrintCategories IS
CURSOR cCat IS
SELECT CATEGORYID, CATEGORYNAME, DESCRIPTION FROM NW_CATEGORIES ;

vCat cCat%ROWTYPE;
BEGIN
OPEN cCat;
   LOOP
      FETCH cCat INTO vCat;
      EXIT WHEN cCat%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(vCat.CATEGORYNAME || ':  '  ||
vCat.DESCRIPTION);
   END LOOP;
   CLOSE cCat;
END;
```

7. Definire un cursore per stampare i 10 clienti più affezionati (per numero di ordini)

10

I cursori parametrici

- Stampa di tutti i prodotti di una categoria (in input)

```
CREATE PROCEDURE PrintProd (vIDCat number) IS
   CURSOR cProd (pCat IN NW_CATEGORIES.CATEGORYID%TYPE) IS
   SELECT *  FROM NW_PRODUCTS WHERE  CATEGORYID=pCat;
BEGIN
FOR vProd IN cProd LOOP
     DBMS_OUTPUT.PUT_LINE(vProd.PRODUCTID || ':  '  ||
vProd.PRODUCTNAME);
  END LOOP;
END;
```

8. Data una città in input, stampare tutti i clienti residenti e per ciascuno la lista dei prodotti ordinati.

11

I cursori for UPDATE

- Aumenta del 10% il prezzo dei prodotti

```
create table NW1_Products as select *  from NW.Products;

CREATE PROCEDURE IncPrice  IS
   CURSOR cProd  IS
   SELECT *  FROM NW1_PRODUCTS FOR UPDATE OF UNITPRICE;
BEGIN
FOR vProd IN cProd LOOP
     UPDATE NW1_PRODUCTS SET UNITPRICE=1.1*UNITPRICE WHERE
CURRENT OF cProd;
END LOOP;
END;

CREATE PROCEDURE IncPrice1  IS
BEGIN
  UPDATE NW1_PRODUCTS SET UNITPRICE=1.1*UNITPRICE;
END;
```

9. Aumenta del P% il prezzo dei prodotti di un fornitore F, se il prodotto è già in riordino l'aumento sarà del P/2%

12

Esercizi

10. Scrivere una procedura che visualizza il nome del cliente associato a un ordine. Gestire il caso di ordine non presente.
11. Scrivere una procedura che stampa i fornitori e l'elenco dei prodotti forniti
12. Scrivere una procedura che restituisca separatamente il conteggio dei prodotti in tre fasce di prezzo date in input.
13. Scrivere una funzione che verifichi che un certo prodotto P sia presente in quantità > Q
14. Definire una tabella di appoggio:
    REORDER(idSupplier, idProduct, quantity, date)
    Scrivere una procedura che inserisce in REORDER un record per ciascun prodotto per cui la quantità è inferiore al livello di riordino

13