

ORACLE

Corso APEX – beanTech

Ing. Roberto Capancioni

Lezione 9 - 10 Ottobre 2023



Chi Sono

Ing. Roberto Capancioni
—Sviluppo Oracle APEX

capancioni.com

 **Oracle ACE
Associate**



Email: sviluppo@capancioni.com
Linkedin: <https://www.linkedin.com/in/robertocapancioni>
Web: <https://capancioni.com>

Trigger

Trigger

Blocco di Codice PL/SQL eseguito all'accadere di uno specifico evento

Quale evento?

- **Istruzione DML** (INSERT UPDATE DELETE su una tabella)
- **Istruzione DDL** (ALTER o CREATE di un oggetto sul database o schema)
- **Evento di Sistema** (startup o shutdown database)
- **Evento utente** (login o logout)

Trigger

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER } triggering_event ON table_name
[FOR EACH ROW]
[FOLLOWS | PRECEDES another_trigger]
[ENABLE / DISABLE ]
[WHEN condition]
DECLARE
    declaration statements
BEGIN
    executable statements
EXCEPTION
    exception_handling statements
END;
```

Trigger

```
create or replace trigger d07_cliente_audit_trg
  after
  update or delete
  on d07_cliente
  for each row
declare
  l_azione varchar2(10);
begin

  l_azione := case
    when updating then 'UPDATE'
    when deleting then 'DELETE'
  end;

  insert into d07_audit (tabella, azione, utente, data)
  values('CLIENTE', l_azione,
  coalesce(sys_context('APEX$SESSION','APP_USER'),user), sysdate);
end;
/
```

Trigger



di **Codice**

Triggering Event	:NEW	:OLD
INSERT	Yes	No
UPDATE	Yes	Yes
DELETE	No	yes

Dog - File

D09 - File

Quick SQL

```
1  persona
2  |
3  | nome vc20
4  | foto file
5  | #prefix:d172
6  |
```

SQL

Impostazioni Scarica Salva script

```
1  -- create tables
2  create table d172_persona (
3      id          number generated by default on null as identity
4      |          |          |          |          |          |          |          |          |
5      | nome      varchar2(20 char),
6      | foto      blob,
7      | foto_filename varchar2(512 char),
8      | foto_mimetype varchar2(512 char),
9      | foto_charset varchar2(512 char),
10     | foto_lastupd date
11 )
12 ;
13
```

Crea un'applicazione



Nome
D172 - File

Aspetto
Vita, Menu laterale



Pagine



Aggiungi pagina



Home

Vuota

Modifica



Persona

Report interattivo con form (d172_persona)

Modifica



D172 Persona



Nome

Foto

Scegli file



D09 - File

Quick SQL

```
1  persona
2  |
3  | nome vc20
4  | foto file
5  |
6  | #prefix:d172
```

SQL

Impostazioni Scarica Salva script

```
1  -- create tables
2  create table d172_persona (
3      id          number generated by default on null as identity
4      |          |          |          |          |          |          |
5      | nome      varchar2(20 char),
6      | foto      blob,
7      | foto_filename varchar2(512 char),
8      | foto_mimetype varchar2(512 char),
9      | foto_charset varchar2(512 char),
10     | foto_lastupd date
11 )
12 ;
13
```

Crea un'applicazione



Nome
D172 - File

Aspetto
Vita, Menu laterale



Pagine ?



Aggiungi pagina



Home

Vuota

Modifica



Persona

Report interattivo con form (d172_persona)

Modifica



D09 - File

D172 Persona

Nome

Foto

Scegli file

Applicazione 661 \ Page Designer

Layout Ricerca pagina Guida

Pagina 3: D172 Persona

- Precedente al rendering
- Componenti
 - Dialog Header
 - Content Body
 - D172 Persona
 - Region Body
 - P3_ID
 - P3_NOME
 - P3_FOTO**
 - P3_FOTO_FILENAME
 - P3_FOTO_MIMETYPE
 - P3_FOTO_CHARSET
 - P3_FOTO_LASTUPD
 - Dialog Footer
 - Pulsanti
 - Close
 - CANCEL
 - Delete
 - DELETE
 - Next
 - SAVE
 - CREATE
- Successivo al rendering

D172 Persona

DIALOG HEADER

CONTENT BODY

D172 Persona

PREVIOUS

REGION BODY

P3_NOME

P3_FOTO

CONTENUTO AREA

SUB REGIONS

NEXT

DIALOG FOOTER

Pulsanti

PREVIOUS

Aree Elementi Pulsanti

Area di testo Cam... Campo di testo A** Campo di testo con com... auto... Campo num...

Gruppo

Elemento

Filtro

- Identificazione
 - Nome
 - P3_FOTO
 - Tipo
 - Sfoglia file...
- Etichetta
 - Etichetta
 - Foto
- Impostazioni
 - Visualizza come
 - Inline File Browse
 - Titolo zona di rilascio
 - Tipo di memoria
 - BLOB column specified in Item :
 - Colonna tipo MIME
 - FOTO_MIMETYPE
 - Colonna nome file
 - FOTO_FILENAME
 - Colonna set di caratteri
 - FOTO_CHARSET
 - Colonna ultimo aggiornamento BLOB
 - FOTO_LASTUPD
 - Visualizza collegamento di download
 - ☒

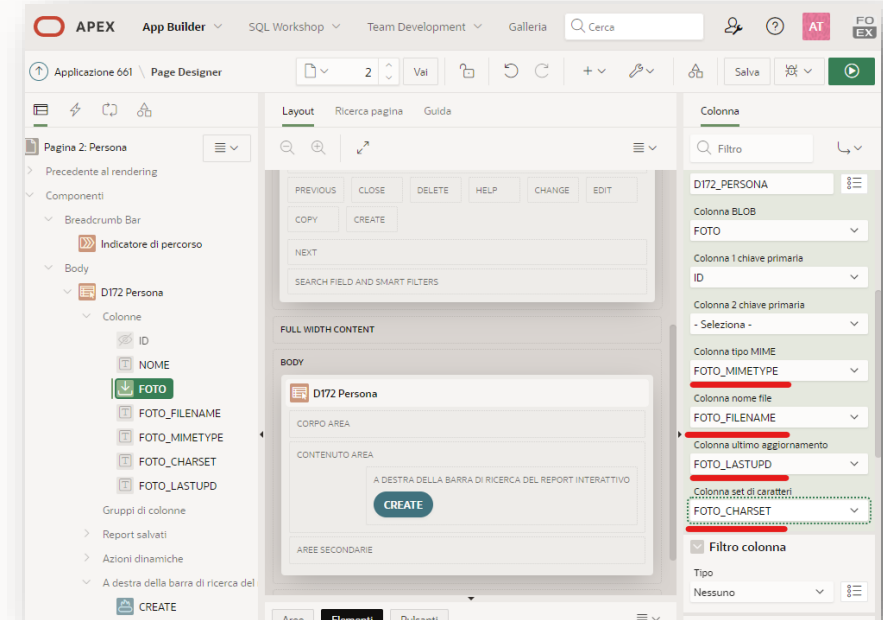


D09 - File

Persona

Q	Vai	Azioni	Crea
	Nome ↑	Foto	
	Logo	Scarica	
1 - 1			

```
select "ID",  
       "NOME",  
       sys.dbms_lob.getlength("FOTO") "FOTO",  
       "FOTO_FILENAME",  
       "FOTO_MIMETYPE",  
       "FOTO_CHARSET",  
       "FOTO_LASTUPD"  
from "D09_PERSONA"
```

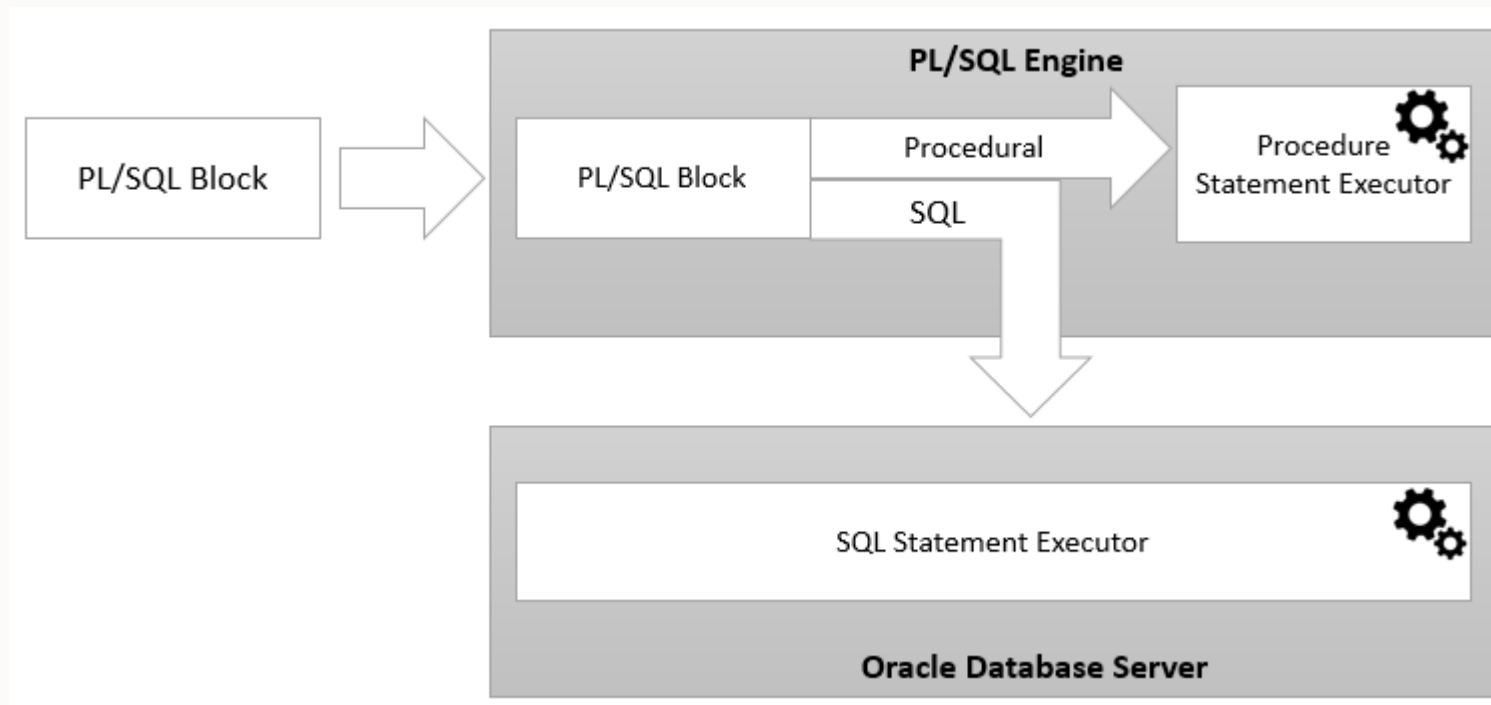


PL/SQL

PL/SQL

- Aggiunge costrutti procedurali al linguaggio SQL
- E' un linguaggio altamente strutturato
- E' integrato all'interno del database Oracle

Procedural
Language extensions to the
Structured
Query
Language



PL/SQL

BEGIN

DBMS_OUTPUT.put_line ('Hello World!');

END;

/

//

PL/SQL

DECLARE

 l_message VARCHAR2(255) := 'Hello World!';

BEGIN

 DBMS_OUTPUT.PUT_LINE(l_message);

END;

/

PL/SQL

DECLARE

V_RESULT NUMBER;

BEGIN

V_RESULT := 1 / 0;

EXCEPTION

WHEN ZERO_DIVIDE THEN

DBMS_OUTPUT.PUT_LINE('Non puoi dividere per zero');

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE(**SQLERRM);**

END;

/

Exception	Oracle Error
ACCESS_INTO_NULL	ORA-06530
CASE_NOT_FOUND	ORA-06592
COLLECTION_IS_NULL	ORA-06531
CURSOR_ALREADY_OPEN	ORA-06511
DUP_VAL_ON_INDEX	ORA-00001
INVALID_CURSOR	ORA-01001
INVALID_NUMBER	ORA-01722
LOGIN_DENIED	ORA-01017
NO_DATA_FOUND	ORA-01403
NOT_LOGGED_ON	ORA-01012
PROGRAM_ERROR	ORA-06501
ROWTYPE_MISMATCH	ORA-06504
SELF_IS_NULL	ORA-30625
STORAGE_ERROR	ORA-06500
SUBSCRIPT_BEYOND_COUNT	ORA-06533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532
SYS_INVALID_ROWID	ORA-01410
TIMEOUT_ON_RESOURCE	ORA-00051
TOO_MANY_ROWS	ORA-01422
VALUE_ERROR	ORA-06502
ZERO_DIVIDE	ORA-01476

PL/SQL

DECLARE

 l_data date;

BEGIN

SELECT sysdate

INTO l_data

FROM dual;

 DBMS_OUTPUT.PUT_LINE(to_char(l_data, 'DD/MM/YYYY HH24:MM:SS'));

END;

/

PL/SQL

DECLARE

 l_data date;

BEGIN

SELECT sysdate

INTO l_data

FROM dual;

 DBMS_OUTPUT.PUT_LINE(to_char(l_data, 'DD/MM/YYYY HH24:MM:SS'));

END;

/

PL/SQL

DECLARE

l_venduto NUMBER := 100000;

BEGIN

IF l_venduto > 100000 **THEN**

DBMS_OUTPUT.PUT_LINE('Venduto > 100k');

ELSIF l_venduto > 50000 **THEN**

DBMS_OUTPUT.PUT_LINE('Venduto > 50k');

ELSE

DBMS_OUTPUT.PUT_LINE('Venduto <= 50k');

END IF;

END;

/

PL/SQL

DECLARE

l_voto CHAR(1);

l_giudizio VARCHAR2(20);

BEGIN

l_voto := 'B';

CASE l_voto

WHEN 'A' THEN

l_giudizio := 'Eccellente' ;

WHEN 'B' THEN

l_giudizio := 'Ottimo' ;

WHEN 'C' THEN

l_giudizio := 'Buono' ;

WHEN 'D' THEN

l_giudizio := 'Sufficiente' ;

ELSE

l_giudizio := 'Insufficiente' ;

END CASE;

DBMS_OUTPUT.PUT_LINE(l_giudizio);

END;

/

PL/SQL DECLARE

```
    l_venduto      NUMBER;
    l_commissione  NUMBER;
BEGIN
    l_venduto := 150000;
    CASE
    WHEN l_venduto > 200000 THEN
        l_commissione := 0.2;
    WHEN l_venduto >= 100000 AND l_venduto < 200000 THEN
        l_commissione := 0.15;
    WHEN l_venduto >= 50000 AND l_venduto < 100000 THEN
        l_commissione := 0.1;
    WHEN l_venduto > 30000 THEN
        l_commissione := 0.05;
    ELSE
        l_commissione := 0;
    END CASE;

    DBMS_OUTPUT.PUT_LINE( 'Commissioni:  ' || l_commissione * 100 || '%' );
END;
/
```

```
DECLARE
```

```
BEGIN
```

```
  for c in (select level numero from dual connect by level <=10)
```

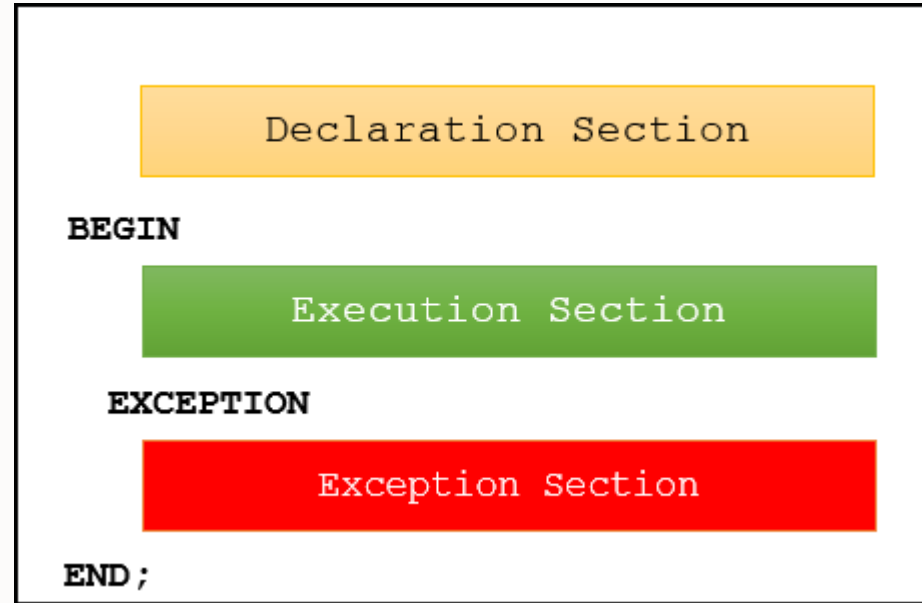
```
  loop
```

```
    DBMS_OUTPUT.PUT_LINE( 'Numero--> ' || c.numero );
```

```
  end loop;
```

```
END;
```

Blocco Anonimo



BLOCCO ANONIMO

```
DECLARE
    l_venduto NUMBER := 100000;
BEGIN
    IF l_venduto > 100000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 100k ' );
    ELSIF l_venduto > 50000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 50k ' );
    ELSE
        DBMS_OUTPUT.PUT_LINE( 'Venduto <= 50k ' );
    END IF;
END;
/
```

PROCEDURA

```
CREATE OR REPLACE PROCEDURE D09_VENDUTO
(
    p_venduto IN number default 0
)
IS
BEGIN
    IF p_venduto > 100000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 100k ' );
    ELSIF p_venduto > 50000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 50k ' );
    ELSE
        DBMS_OUTPUT.PUT_LINE( 'Venduto <= 50k ' );
    END IF;
END;
/

BEGIN
    D09_VENDUTO(p_venduto => 100000);
END;
/
```

BLOCCO ANONIMO

```

DECLARE
    l_venduto NUMBER := 100000;
BEGIN
    IF l_venduto > 100000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 100k ' );
    ELSIF l_venduto > 50000 THEN
        DBMS_OUTPUT.PUT_LINE( 'Venduto > 50k ' );
    ELSE
        DBMS_OUTPUT.PUT_LINE( 'Venduto <= 50k ' );
    END IF;
END;
/

```

FUNZIONE

```

CREATE OR REPLACE FUNCTION D09_GET_VENDUTO
(
    p_venduto IN number default 0
)
RETURN VARCHAR2
IS
    l_return VARCHAR2(100);
BEGIN
    IF p_venduto > 100000 THEN
        l_return := 'Venduto > 100k ';
    ELSIF p_venduto > 50000 THEN
        l_return := 'Venduto > 50k ';
    ELSE
        l_return := 'Venduto <= 50k ';
    END IF;
    RETURN l_return;
END;
/

DECLARE
    l_venduto VARCHAR2(100) := null;
BEGIN
    l_venduto := D09_GET_VENDUTO( p_venduto => 100000 );
    DBMS_OUTPUT.PUT_LINE( l_venduto );
END;
/

```

PL/SQL

PROCEDURE E FUNZIONI

```
CREATE OR REPLACE PROCEDURE D09_VENDUTO ...  
...
```

```
CREATE OR REPLACE FUNCTION D09_GET_VENDUTO ...  
...
```

PACKAGE

```
CREATE OR REPLACE PACKAGE D09_PKG  
IS  
  PROCEDURE VENDUTO (  
    p_venduto IN number default 0  
  );  
  
  FUNCTION GET_VENDUTO (  
    p_venduto IN number default 0  
  )  
  RETURN VARCHAR2;  
  
END D09_PKG;  
/
```

```

CREATE OR REPLACE PACKAGE D09_PKG
IS
PROCEDURE VENDUTO (
    p_venduto IN number default 0
);

FUNCTION GET_VENDUTO (
    p_venduto IN number default 0
)
RETURN VARCHAR2;

END D09_PKG;
/

```

```

CREATE OR REPLACE PACKAGE BODY D06_PKG
IS
    PROCEDURE VENDUTO (
        p_venduto IN number default 0
    )
    IS
    BEGIN
        IF p_venduto > 100000 THEN
            DBMS_OUTPUT.PUT_LINE( 'Venduto > 100k ' );
        ELSIF p_venduto > 50000 THEN
            DBMS_OUTPUT.PUT_LINE( 'Venduto > 50k ' );
        ELSE
            DBMS_OUTPUT.PUT_LINE( 'Venduto <= 50k ' );
        END IF;
    END VENDUTO;

    FUNCTION GET_VENDUTO (
        p_venduto IN number default 0
    )
    RETURN VARCHAR2
    IS
        l_return VARCHAR2(100);
    BEGIN
        IF p_venduto > 100000 THEN
            l_return := 'Venduto > 100k ';
        ELSIF p_venduto > 50000 THEN
            l_return := 'Venduto > 50k ';
        ELSE
            l_return := 'Venduto <= 50k ';
        END IF;
        RETURN l_return;
    END GET_VENDUTO;
END D06_PKG;
/

```

Richiamo Procedura nel Package

```
BEGIN  
    D09_PKG.VENDUTO(p_venduto => 100000);  
END;  
/
```

Richiamo Funzione nel Package

```
DECLARE  
    l_venduto VARCHAR2(100) := null;  
BEGIN  
    l_venduto := D09_PKG.GET_VENDUTO(p_venduto => 100000);  
    DBMS_OUTPUT.PUT_LINE( l_venduto );  
END;  
/
```