

Corso APEX – Italia Marittima

- **Ing. Roberto Capancioni**
- **Lezione 4 - 28 Settembre 2023**

Chi Sono

Ing. Roberto Capancioni
—Sviluppo Oracle APEX

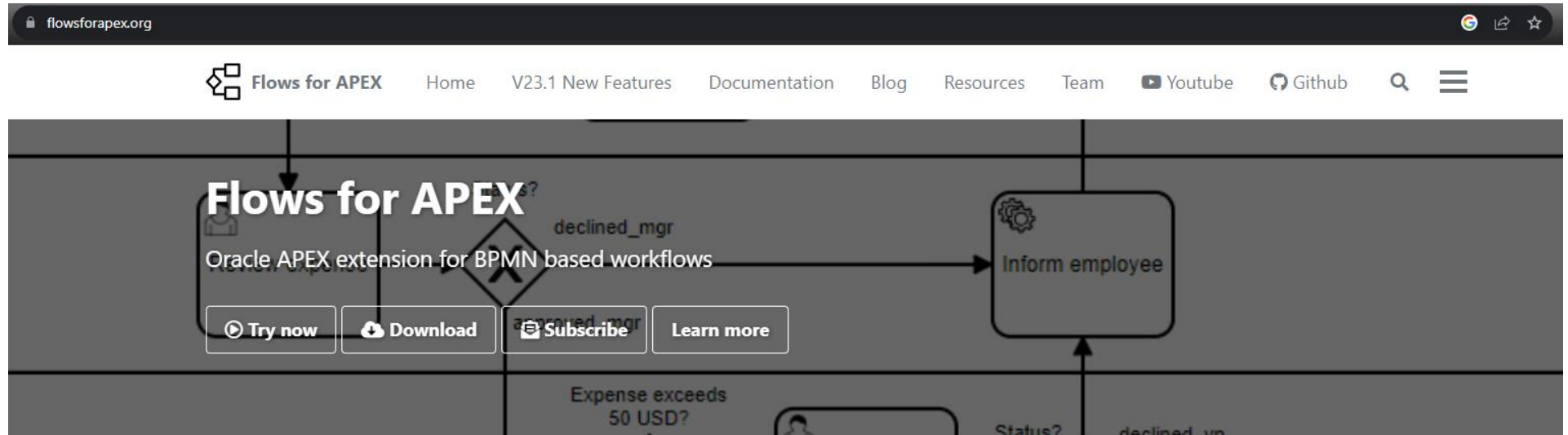
capancioni.com



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

Flow for APEX

Flows for APEX



Flows for APEX is a powerful workflow engine extension for Oracle APEX applications. Model your business processes with BPMN, develop your process steps in APEX, and monitor your running processes with the Flows for APEX application. Flows for APEX was created as a community project by team of experienced APEX professionals to meet the workflow requirements in many APEX applications.



Watch "What's New in Flows for APEX v23.1" on APEX Office Hours

The Flows for APEX team showed some of the v23.1 new features in our recent Oracle APEX Office Hours session: What's New in Flows for APEX v 23.1. See demos for:

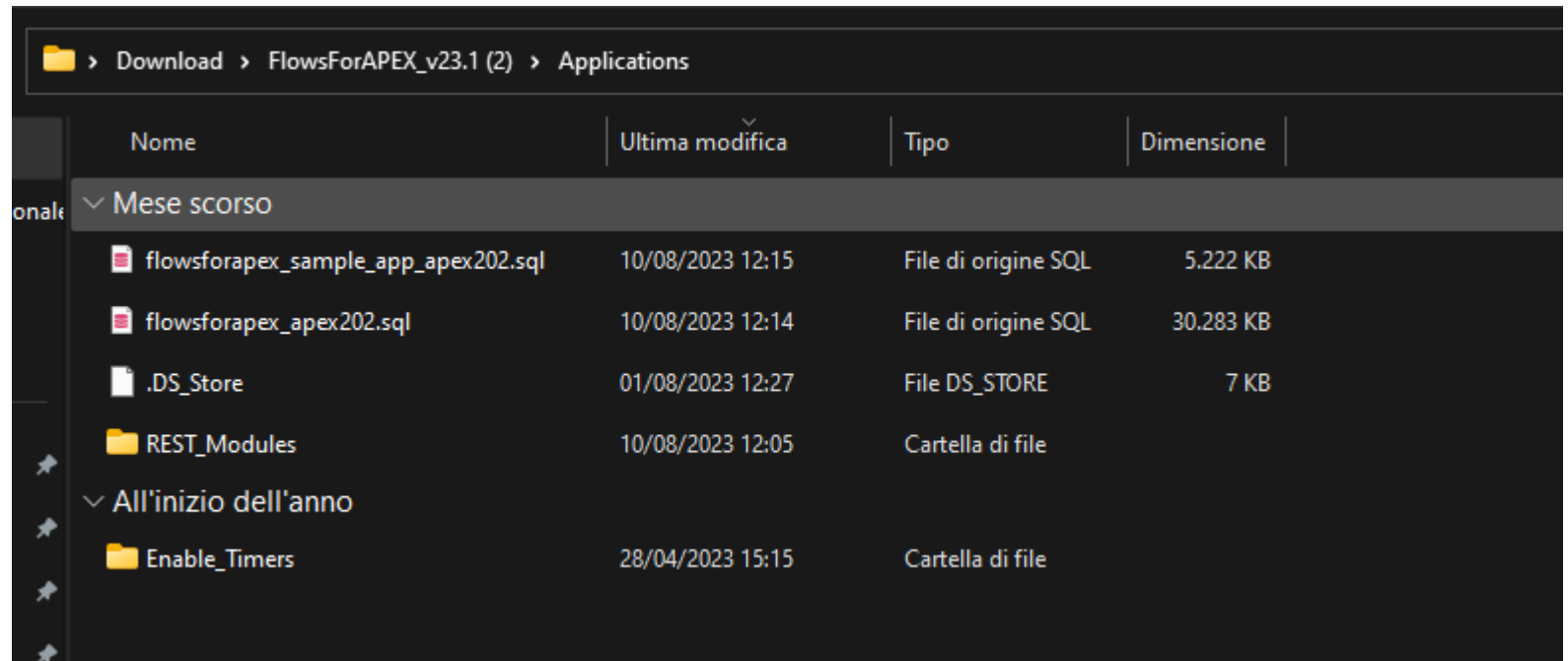
- Inter-process communication with BPMN Message Flow
- REST interface
- Task Assignment, Due Dates and Priorities
- Timestamp Process Variables and Binding Everywhere
- Updated Process Modeler and Viewer
- System Administration Upgrades with Diagram Logging, Instance Summaries, and Statistics

Flows for APEX

- `flowsforapex_apex202.sql`
- `flowsforapex_sample_app_apex202.sql`

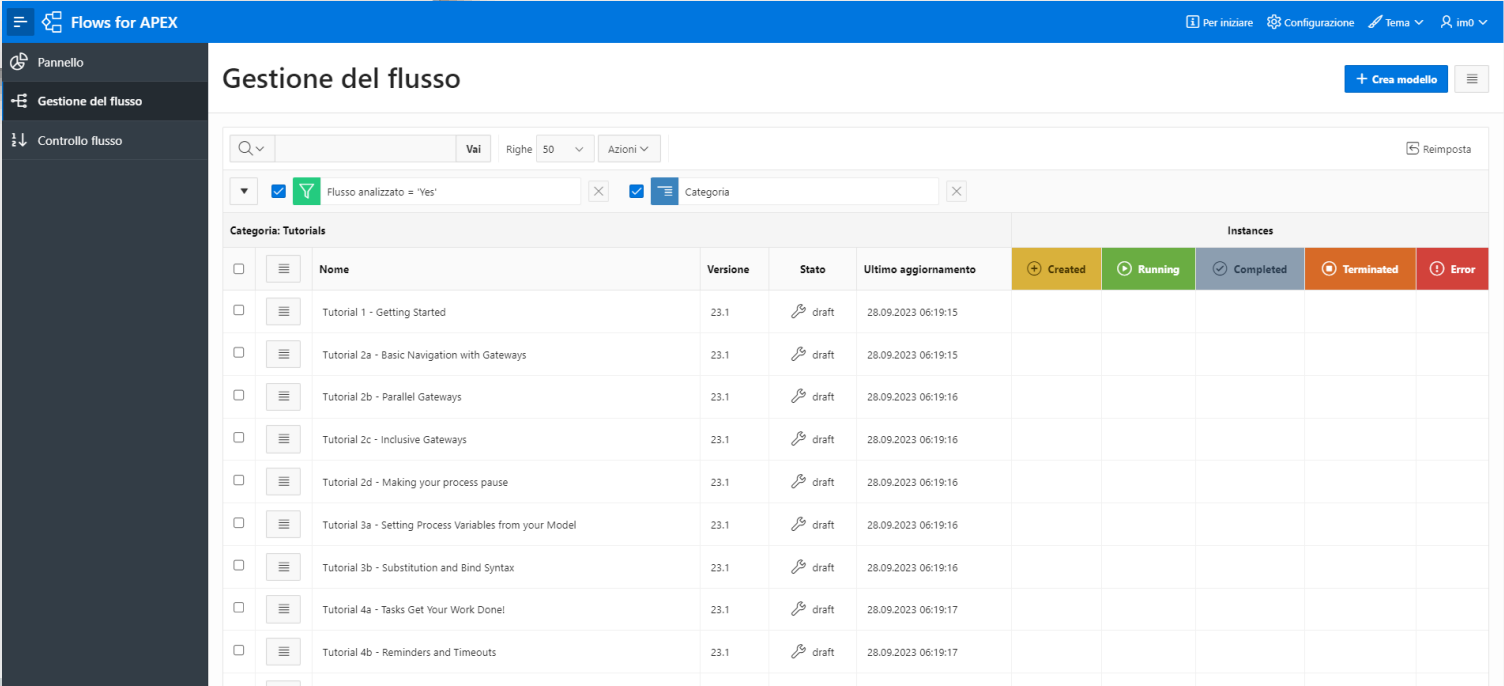
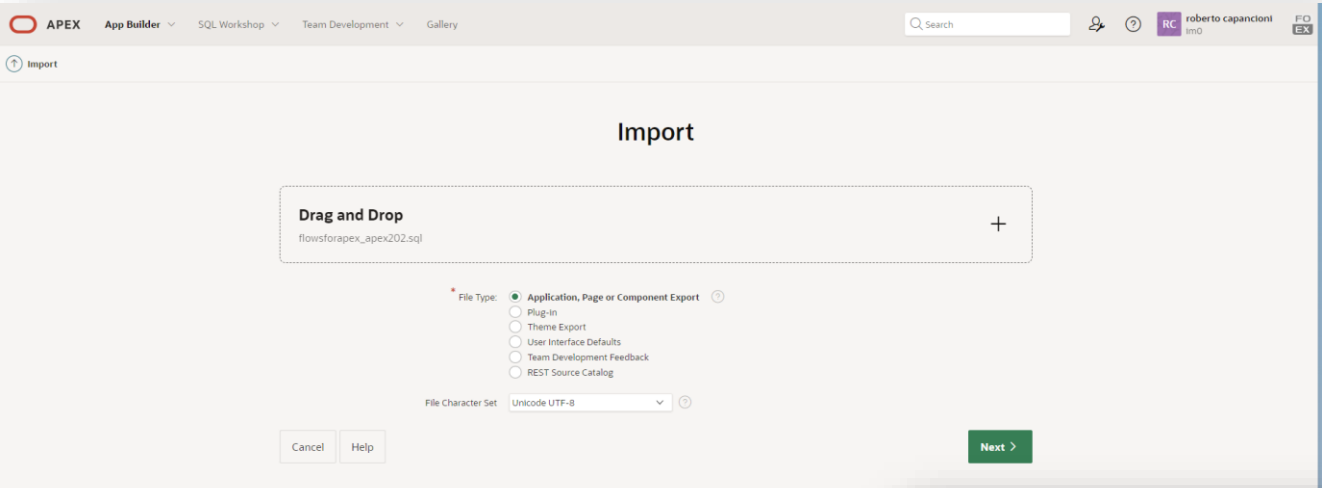
← Applicazione

← Esempi



	Nome	Ultima modifica	Tipo	Dimensione
▼	Mese scorso			
	flowsforapex_sample_app_apex202.sql	10/08/2023 12:15	File di origine SQL	5.222 KB
	flowsforapex_apex202.sql	10/08/2023 12:14	File di origine SQL	30.283 KB
	.DS_Store	01/08/2023 12:27	File DS_STORE	7 KB
	REST_Modules	10/08/2023 12:05	Cartella di file	
▼	All'inizio dell'anno			
	Enable_Timers	28/04/2023 15:15	Cartella di file	

Flows for APEX



Flows for APEX

The screenshot shows the APEX web interface with the 'Import' dialog box open. The dialog has a title bar with 'Import' and an upward arrow icon. The main area is titled 'Import' and contains a 'Drag and Drop' section with a dashed border and a plus sign icon. Below this, there is a 'File Type' section with a red asterisk and a help icon. The 'File Type' section has five radio button options: 'Application, Page or Component Export' (selected), 'Plug-in', 'Theme Export', 'User Interface Defaults', and 'Team Development Feedback'. Below the radio buttons is a 'File Character Set' section with a dropdown menu set to 'Unicode UTF-8' and a help icon. At the bottom of the dialog are three buttons: 'Cancel', 'Help', and 'Next >'.

Import

Drag and Drop
flowsforapex_sample_app_apex202.sql

* File Type: ☒ Application, Page or Component Export ?
☐ Plug-in
☐ Theme Export
☐ User Interface Defaults
☐ Team Development Feedback
☐ REST Source Catalog

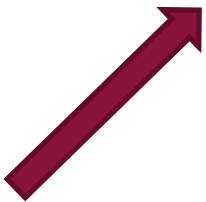
File Character Set: Unicode UTF-8

Cancel Help Next >

Merge

D18 MERGE

```
merge into d18_vendita d
  using d17_vendita s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
      )
when matched then
  update set
    --d.data      = s.data,
    --d.prodotto  = s.prodotto,
    --d.cliente   = s.cliente,
    d.zona_cliente = s.zona_cliente,
    d.quantita    = s.quantita,
    d.importo     = s.importo
when not matched then
  insert (d.data,d.prodotto,d.cliente,d.zona_cliente,d.quantita,d.importo)
  values (s.data,s.prodotto,s.cliente,s.zona_cliente,s.quantita,s.importo);
```



Non si possono aggiornare le colonne presenti nella clausola on

D18 MERGE

```
merge into d18_vendita d
  using d17_vendita s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
    )
```

```
merge into d18_vendita d
  using (select *
        from d17_vendita) s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
    )
```

```
merge into d18_vendita d
  using (select
        trunc(sysdate) data,
        'Pane' prodotto,
        'Mario Rossi' cliente,
        'NORD' zona_cliente,
        100 quantita,
        400 importo
        from dual) s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
    )
```

D18 MERGE

```
merge into d18_vendita d
  using d17_vendita s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
      )
  when matched then
    update set
      d.zona_cliente = s.zona_cliente,
      d.quantita     = s.quantita,
      d.importo      = s.importo
    where d.importo < 100
  when not matched then
    insert (d.data,d.prodotto,d.cliente,d.zona_cliente,d.quantita,d.importo)
    values (s.data,s.prodotto,s.cliente,s.zona_cliente,s.quantita,s.importo) ;
```

D18 MERGE

```
merge into d18_vendita d
  using d17_vendita s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
      )
when matched then
  update set
    d.zona_cliente = s.zona_cliente,
    d.quantita     = s.quantita,
    d.importo      = s.importo
  where d.importo < 100
delete where d.importo > 900
when not matched then
  insert (d.data,d.prodotto,d.cliente,d.zona_cliente,d.quantita,d.importo)
  values (s.data,s.prodotto,s.cliente,s.zona_cliente,s.quantita,s.importo) ;
```

D18 MERGE

```
merge into d18_vendita d
  using d17_vendita s
    on (      s.data      = d.data
        and s.prodotto = d.prodotto
        and s.cliente  = d.cliente
      )
when matched then
  update set
    d.zona_cliente = s.zona_cliente,
    d.quantita     = s.quantita,
    d.importo      = s.importo
  where d.importo < 100
delete where d.importo > 900
when not matched then
  insert (d.data,d.prodotto,d.cliente,d.zona_cliente,d.quantita,d.importo)
  values (s.data,s.prodotto,s.cliente,s.zona_cliente,s.quantita,s.importo)
  where s.importo < 100;
```


D18 MERGE

```
create unique index d18_vendita_uq1  
on d18_vendita (data,prodotto,cliente);
```

```
begin  
    DBMS_STATS.GATHER_TABLE_STATS(null, 'D08_VENDITA');  
    DBMS_STATS.GATHER_TABLE_STATS(null, 'D18_VENDITA');  
end;
```

D303 Dipendente Attività

D303 Dipendente Attività

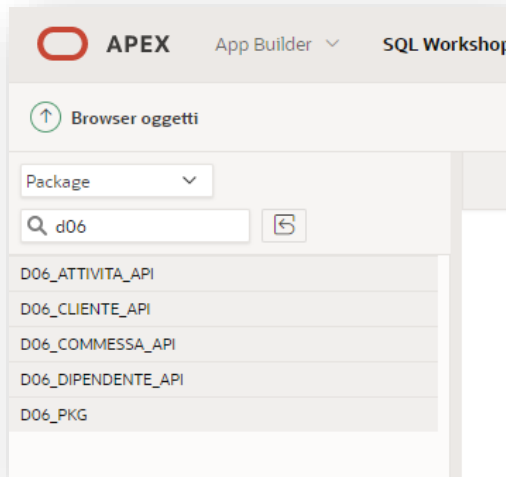
 **APEX** App Builder ▾ SQL Workshop ▾ Team Development ▾ Gallery

↑ Utilities \ Quick SQL

Quick SQL Clear Load Help Save Model **Generate SQL**

```
1 dipendente /insert 5
2   dipendente /nn /values francesca, marco, anna, simone, paola /unique
3   tariffa num /nn /values 200,150
4
5 cliente /insert 5
6   cliente vc50 /nn /values cliente_01,cliente_02,cliente_03,cliente_04,cliente_05 /unique
7
8 commessa /insert 5
9   commessa vc30 /nn /values COMM01, COMM02, COMM03, COMM04, COMM05 /unique
10
11 attivita /insert 200
12   dipendente_id /nn
13   cliente_id /nn
14   commessa_id vc100 /nn
15   data d /nn
16   ore num /nn
17
18 # settings = { prefix: "D06", ondelete: "RESTRICT", api: true, semantics: "CHAR", language: "EN", APEX: true }
```

D303 Dipendente Attività



```
create or replace package d06_cliente_api
is
    /* example:
    declare
        l_cliente          varchar2(10);
    begin
        d06_cliente_api.get_row (
            p_id             => 1,
            p_cliente        => l_cliente
        );
    end;
    */
    procedure get_row (
        p_id                in number,
        p_cliente           out varchar2
    );

    /* example:
    begin
        d06_cliente_api.insert_row (
            p_id             => null,
            p_cliente        => null
        );
    end;
    */
    procedure insert_row (
        p_id                in number default null,
        p_cliente           in varchar2 default null
    );
    procedure update_row (
        p_id                in number default null,
        p_cliente           in varchar2 default null
    );
    procedure delete_row (
        p_id                in number
    );
end d06_cliente_api;
```

```
create or replace package body d06_cliente_api
is
    procedure get_row (
        p_id                in number,
        p_cliente           out varchar2
    )
    is
    begin
        for c1 in (select * from d06_cliente where id = p_id) loop
            p_cliente := c1.cliente;
        end loop;
    end get_row;

    procedure insert_row (
        p_id                in number default null,
        p_cliente           in varchar2 default null
    )
    is
    begin
        insert into d06_cliente (
            id,
            cliente
        ) values (
            p_id,
            p_cliente
        );
    end insert_row;
    procedure update_row (
        p_id                in number default null,
        p_cliente           in varchar2 default null
    )
    is
    begin
        update d06_cliente set
            id = p_id,
            cliente = p_cliente
        where id = p_id;
    end update_row;
    procedure delete_row (
        p_id                in number
    )
    is
    begin
        delete from d06_cliente where id = p_id;
    end delete_row;
end d06_cliente_api;
```

D303 Dipendente Attività

Crea un'applicazione



Nome
D06 Dipendente Attività

Aspetto
Vita, Menu laterale



Pagine



Aggiungi pagina



Home

Vuota

Modifica



Dipendente

Report interattivo con form (d06_dipendente)

Modifica



Cliente

Report interattivo con form (d06_cliente)



Commissa

Report interattivo con form (d06_commissa)



Attività

Report interattivo con form (d06_attività)

Funzioni

Seleziona tutto



Installa applicazione Web
progressiva



Pagina delle informazioni

Aggiunge una pagina di informazioni su
questo applicazione.



Annulla

Crea applicazione

D06 Dipendente Attività

- Home
- Dipendente
- Cliente
- Commissa
- Attività



D06 Dipendente Attività

Dipendente

Cliente

Commissa

Attività

D303 Dipendente Attività

The screenshot displays the Oracle APEX App Builder interface. The main workspace shows a form titled 'Cliente' with a 'DIALOG HEADER' and a 'CONTENT BODY'. The 'CONTENT BODY' contains a 'REGION BODY' with a text input field labeled 'P5_CLIENTE'. The 'Process' pane on the right shows a flow starting with 'Elabora form Cliente', followed by 'Inserisci da Codice', and 'Chiudi finestra di dialogo'. The 'Condition' pane shows a 'Condizione lato server' with the condition 'Se si fa clic sul pulsante' set to 'Mai'.

/* example:

```
begin
  d06_cliente_api.insert_row (
    p_id
    p_cliente
  );
end;
```

*/

=> nul
=> nul

begin

```
d06_cliente_api.insert_row (
  p_cliente => :P5_CLIENTE
);
```

end;

Condizione lato server

Se si fa clic sul
pulsante

CREATE

Tipo

- Seleziona -

D303 Dipendente Attività

The screenshot shows the Oracle APEX App Builder interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Galleria' tabs. A search bar is on the right. The main workspace is divided into three panes: 'Page Designer' on the left, 'Layout' in the center, and 'Process Designer' on the right. The 'Page Designer' pane shows a process flow with steps: 'Dopo la sottomissione', 'Convalida in corso', 'Elaborazione' (containing 'Elabora form Cliente', 'Inserisci da Codice', 'Modifica da Codice' (highlighted), and 'Chiudi finestra di dialogo'), 'Dopo l'elaborazione', and 'Callback Ajax'. The 'Layout' pane shows a visual representation of the page with a 'SUB REGIONS' section and a 'DIALOG FOOTER' section containing 'Pulsanti' (Previous, Close, Delete, Cancel, Delete buttons). The 'Process Designer' pane shows the configuration for the 'Modifica da Codice' process. It includes fields for 'Nome' (Modifica da Codice), 'Tipo' (Esegui codice), 'Area modificabile' (- Seleziona -), 'Origine' (Database locale, PL/SQL), and a 'Condizione lato server' (Se si fa clic sul pulsante, SAVE). The 'Codice PL/SQL' section contains the following code:

```
/* example:
begin
  d06_cliente_api.insert_row (
    p_id
    p_cliente
  );
end;
*/
```

```
begin
  d06_cliente_api.update_row (
    p_id      => :P5_ID
    p_cliente => :P5_CLIENTE
  );
end;
```

Condizione lato server

Se si fa clic sul pulsante: SAVE

Tipo: - Seleziona -

D303 Dipendente Attività

The screenshot displays the Oracle APEX App Builder interface for 'Applicazione 762' in 'Page Designer' mode. The left sidebar shows the process flow: 'Dopo la sottomissione' -> 'Convalida in corso' -> 'Elaborazione' -> 'Processi' -> 'Elabora form Cliente' -> 'Inserisci da Codice' -> 'Modifica da Codice' (highlighted) -> 'Chiudi finestra di dialogo'. The main canvas shows a 'DIALOG FOOTER' with buttons for 'PREVIOUS', 'CLOSE', 'DELETE', 'CANCEL', and 'DELETE'. The right sidebar shows the configuration for the 'Modifica da Codice' process:

- Identificazione:** Nome (Modifica da Codice), Tipo (Esegui codice), Area modificabile (- Seleziona -).
- Origine:** Posizione (Database locale), Linguaggio (PL/SQL), Codice PL/SQL (begin d06_cliente_api.update_row (p_id => :P5_ID, p_cliente => :P5_CLIENTE); end;).
- Condizione lato server:** Se si fa clic sul pulsante (SAVE), Tipo (- Seleziona -).

```
/* example:
begin
  d06_cliente_api.insert_row (
    p_id
    p_cliente
  );
end;
*/
```

```
begin
  d06_cliente_api.update_row (
    p_id      => :P5_ID
    p_cliente => :P5_CLIENTE
  );
end;
```

D303 Dipendente Attività

The screenshot shows the Oracle APEX Page Designer interface for 'Applicazione 762' in 'Page Designer' mode. The left pane shows the process flow: 'Dopo la sottomissione' (Convalida in corso), 'Elaborazione' (Processi), and 'Dopo l'elaborazione' (Callback Ajax). The 'Processi' list includes 'Elabora form Cliente', 'Inserisci da Codice', 'Modifica da Codice', 'Elimina da Codice' (highlighted), and 'Chiudi finestra di dialogo'. The main canvas shows a 'Cliente' dialog box with a 'DIALOG HEADER' and a 'CONTENT BODY'. The 'CONTENT BODY' contains a 'REGION BODY' with a text input field labeled 'P5_CLIENTE'. The right pane shows the 'Processo' configuration for the 'Elimina da Codice' process. It includes fields for 'Nome' (Elimina da Codice), 'Tipo' (Esegui codice), 'Area modificabile' (- Seleziona -), 'Posizione' (Database locale), 'Linguaggio' (PL/SQL), and 'Codice PL/SQL' (begin d06_cliente_api.delete_row (p_id => :P5_ID); end;). A 'Condizione lato server' section is also visible, showing 'Se si fa clic sul pulsante' (DELETE) and 'Tipo' (- Seleziona -).

/* example:

```
begin
  d06_cliente_api.insert_row (
    p_id
    p_cliente
  );
end;
```

*/

=> nul

=> nul

begin

```
d06_cliente_api.delete_row (
  p_id      => :P5_ID
);
```

end;

Condizione lato server

Se si fa clic sul pulsante

DELETE

Tipo

- Seleziona -

D303 Dipendente Attività

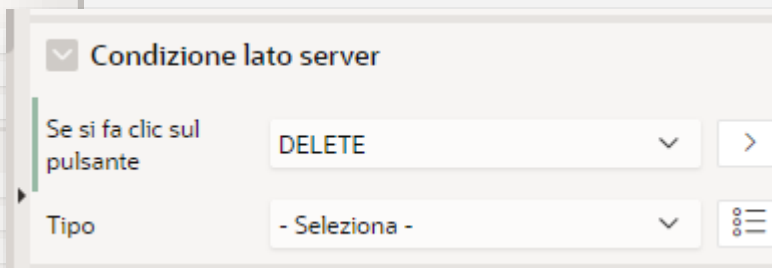
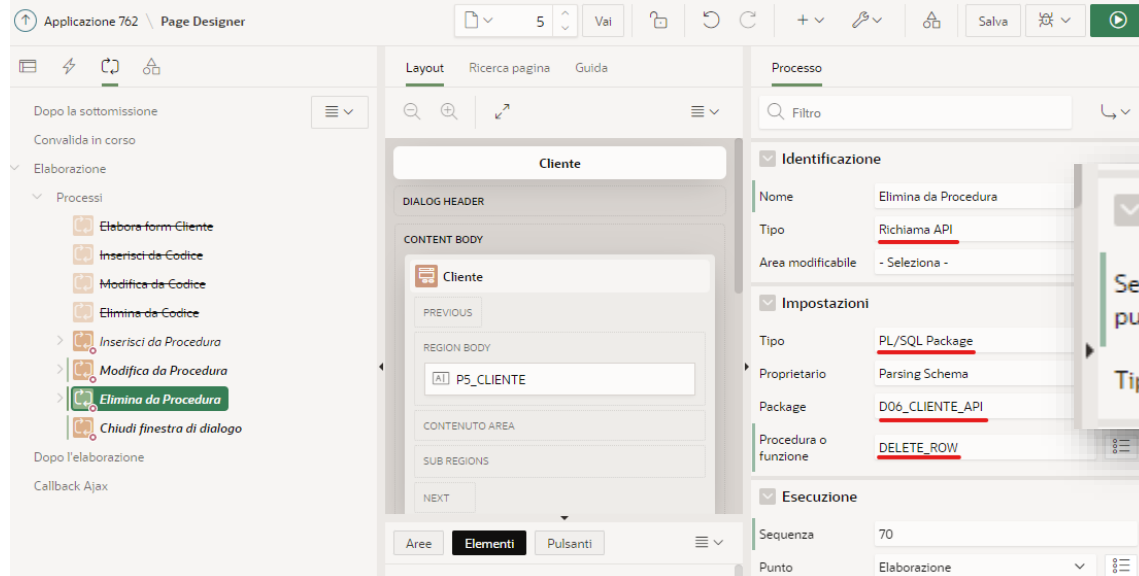
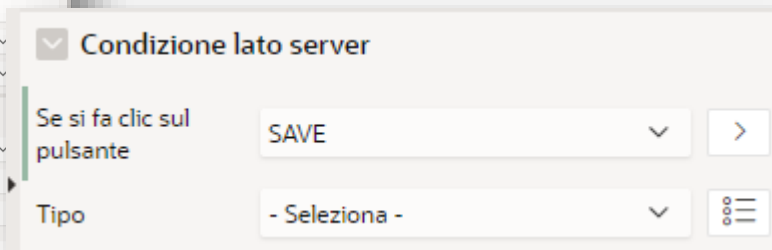
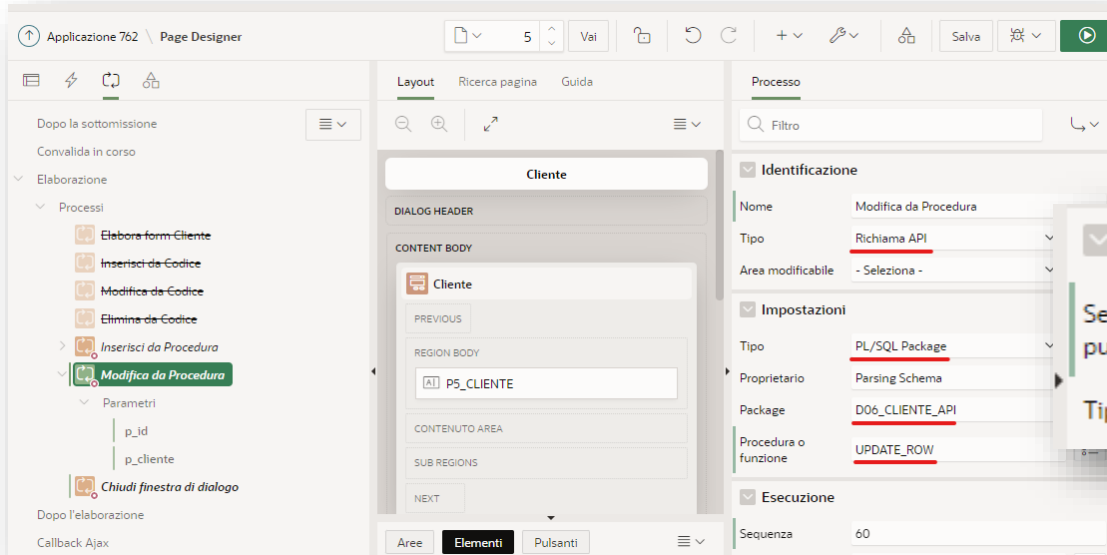
The image displays three overlapping screenshots of the Oracle APEX Page Designer interface, illustrating the configuration of a process step within a page.

Top Left Screenshot: Shows the 'Processo' (Process) tab. The 'Identificazione' (Identification) section is expanded, showing the process name 'Inserisci da Procedura', type 'Richiama API', and area 'Seleziona'. The 'Impostazioni' (Settings) section shows the type 'PL/SQL Package', owner 'Parsing Schema', package 'D06_CLIENTE_API', and procedure 'INSERT_ROW'. The 'Esecuzione' (Execution) section shows the sequence '50' and point 'Elaborazione'.

Top Right Screenshot: Shows the 'Parametro' (Parameter) tab. The 'Identificazione' section shows the parameter name 'p_cliente'. The 'Parametro' section shows the direction 'In', data type 'VARCHAR2', and the value 'P5_CLIENTE'.

Bottom Screenshot: Shows the 'Condizione lato server' (Server-side condition) dialog. It displays the condition 'Se si fa clic sul pulsante' (If the button is clicked) with the value 'CREATE' and the type 'Seleziona'.

D303 Dipendente Attività



D304 Viste

Viste

```
create or replace view d08_acquisto_vw as
select a.id,
       a.fornitore_id,
       f.fornitore,
       a.prodotto_id,
       p.prodotto,
       p.tipo_prodotto_id,
       tp.tipo_prodotto,
       a.data,
       a.quantita
from d08_acquisto a
join d08_fornitore f      on a.fornitore_id      = f.id
join d08_prodotto p       on a.prodotto_id        = p.id
join d08_tipo_prodotto tp on p.tipo_prodotto_id = tp.id
```

Viste

```
create or replace view d08_vendita_vw as
select v.id,
       v.cliente_id,
       c.cliente,
       v.prodotto_id,
       p.prodotto,
       p.tipo_prodotto_id,
       tp.tipo_prodotto,
       v.data,
       v.quantita
from d08_vendita v
join d08_cliente c      on v.cliente_id      = c.id
join d08_prodotto p     on v.prodotto_id     = p.id
join d08_tipo_prodotto tp on p.tipo_prodotto_id = tp.id
```

Viste

```
select v.id,  
       v.cliente_id,  
       c.cliente,  
       v.prodotto_id,  
       p.prodotto,  
       p.tipo_prodotto_id,  
       tp.tipo_prodotto,  
       v.data,  
       v.quantita  
from d08_vendita v  
join d08_cliente c      on v.cliente_id      = c.id  
join d08_prodotto p     on v.prodotto_id     = p.id  
join d08_tipo_prodotto tp on p.tipo_prodotto_id = tp.id  
where v.cliente_id = 3
```

Che Differenza C'è?

```
select *  
  from d08_vendita_vw  
 where cliente_id = 3
```

Viste

```
select v.id,  
       v.cliente_id,  
       c.cliente,  
       v.prodotto_id,  
       p.prodotto,  
       p.tipo_prodotto_id,  
       tp.tipo_prodotto,  
       v.data,  
       v.quantita  
from d08_vendita v  
join d08_cliente c      on v.cliente_id      = c.id  
join d08_prodotto p     on v.prodotto_id     = p.id  
join d08_tipo_prodotto tp on p.tipo_prodotto_id = tp.id  
where v.cliente_id = 3
```

Che Differenza C'è?

```
select *  
  from d08_vendita_vw  
 where cliente_id = 3
```

Filtra solo le vendite del cliente_id 3
SOLO DOPO aver caricato tutte le
vendite

Performance peggiori!!!

Filtra solo le vendite del
cliente_id 3

D304 SQL Macro

SQL Macro

- E' una funzione che restituisce **IL TESTO** di una query SQL
- Tale testo viene compilato come una query **SQL nativa**
- Scrivo PLSQL solo per generare il testo ma poi eseguo codice SQL
- Unisce i **vantaggi** di avere una **Funzione** con performance di una **SQL nativa**

SQL Macro

- **Scalar Expression (equivalente di una colonna calcolata)**
- **Table Expression (query SQL completa)**

SQL Macro

Espressione Scalare

```
create or replace function show_date(p_value date)
return varchar2 sql_macro(scalar)
is
begin
    return q'{ to_char(p_value, 'YYYY-MM-DD') }';
end;
/
```

SQL Macro

Espressione Tabellare

```
create or replace function sal_by_dept
  return varchar2 sql_macro(table) -- (table) può essere omissso
is
begin
  return q'{
    select deptno, sum(sal) as sal_tot
    from    emp
    group by deptno
  }';
end;
/
```

PKG

```
create or replace package d08_pkg is
    function get_vendita (
        p_cliente_id    in number default null,
        p_prodotto_id    in number default null
    ) return clob sql_macro;

    function get_acquisto (
        p_fornitore_id in number default null,
        p_prodotto_id  in number default null
    ) return clob sql_macro;

end d08_pkg;
/
create or replace package body d08_pkg is
    -- CODICE
end d08_pkg;
/
```

PKG

```
create or replace package body d08_pkg is
  function get_vendita (
    p_cliente_id   in number default null,
    p_prodotto_id  in number default null
  ) return clob sql_macro
  is
begin
  RETURN q'{
    select v.id,
           v.cliente_id,
           c.cliente,
           v.prodotto_id,
           p.prodotto,
           p.tipo_prodotto_id,
           tp.tipo_prodotto,
           v.data,
           v.quantita
    from d08_vendita v
    join d08_cliente c on v.cliente_id = c.id
    join d08_prodotto p on v.prodotto_id = p.id
    join d08_tipo_prodotto tp on p.tipo_prodotto_id = tp.id
    where v.cliente_id = nvl(p_cliente_id,v.cliente_id)
           and v.prodotto_id = nvl(p_prodotto_id,v.prodotto_id)
  }';
end get_vendita;
end d08_pkg;
/
```


Query con SQL Macro

```
select * from d08_pkg.get_vendita();  
select * from d08_pkg.get_vendita(p_cliente_id => 3);  
select * from d08_pkg.get_vendita(p_prodotto_id => 68);  
select * from d08_pkg.get_vendita(p_cliente_id => 3,  
                                   p_prodotto_id => 68);
```

```
select * from d08_pkg.get_acquisto();  
select * from d08_pkg.get_acquisto(p_fornitore_id => 8);  
select * from d08_pkg.get_acquisto(p_prodotto_id => 68);  
select * from d08_pkg.get_acquisto(p_fornitore_id => 8,  
                                   p_prodotto_id => 68);
```