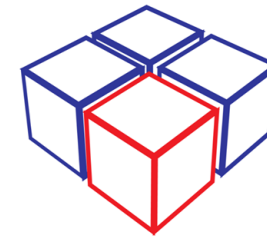


capancioni.com



iema
Software Architectures

Oracle APEX Developing Database Web Applications



23 Maggio 2023

ORACLE

Partner



capancioni.com

Chi Siamo

Ing. Roberto Capancioni

—Sviluppo Oracle APEX



—Board Member ITOUG



Email: sviluppo@capancioni.com

Linkedin: <https://www.linkedin.com/in/robertocapancioni>

ITOUG: <https://itoug.it>

Chi Siamo


- Dott.ssa Nicoletta Bindi
- in Tema dal 2001
- Analista sviluppatrice java e web
- Apex dal 2020



D17 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
predefinite 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, 'D17_VENDITA');  
    DBMS_STATS.GATHER_TABLE_STATS(null, 'D18_VENDITA');  
end;
```

Automazioni

ORACLE

Partner



IEMA
Software Architectures

capancioni.com



Automazioni

APEX

App Builder

SQL Workshop

Team Development

Galleria

Cerca

AT at0

FO EX

↑ Applicazione 766 \ Componenti condivisi

Logica applicazione

Definizione di applicazione

Elementi di applicazione

Processi di applicazione

Calcoli di applicazione

Impostazioni applicazione

Opzioni di creazione

1

Sicurezza

Attributi di sicurezza

Schemi di autenticazione

Schemi di autorizzazione

Controllo dell'accesso applicazione

Protezione stato sessione

1

1

Altri componenti

Liste di valori

Plugin

Impostazioni componente

Scorciatoie

3

Informazioni

È possibile visualizzare o applicare i componenti condivisi in qualsiasi pagina all'interno dell'applicazione. Questa pagina contiene una lista dei componenti condivisi disponibili.

Ulteriori informazioni...

Task

Esporta componenti applicazione

Gestisci oggetti di supporto

Modifica commento applicazione

Commenti sviluppatori

Oggetti area di lavoro

Gli oggetti di area di lavoro vengono memorizzati a livello di area di lavoro, pertanto sono visibili in tutte le applicazioni.

Credenziali

Server remoti

Navigazione e ricerca

Liste

Menu di navigazione

Indicatori di percorso

Lista barra di navigazione

Configurazioni di ricerca

3

1

Interfaccia utente

Attributi interfaccia utente

Applicazione Web progressiva

Temi

Modelli

Modelli di posta elettronica

1

69

File e report

File applicazione statici

File area di lavoro statici

Query report

Layout report

3

1

Origini dati

Definizioni caricamento dati

SQL abilitato per REST

Origini dati REST

Sincronizzazione REST

>

Workflow e automazioni

Definizioni task

Automazioni

Globalizzazione


Attributi di globalizzazione

Messaggi di testo

Traduzioni applicazione

ORACLE

Partner

 **IEMA**
Software Architectures

capancioni.com

Automazioni

Crea automazione

* Nome ?

Tipo ?

Azioni avviate in ?

Pianificazione esecuzione ?

Tipo

Le automazioni con una **pianificazione** verranno eseguite automaticamente in base alla frequenza selezionata. Se è selezionata l'opzione **Su richiesta**, l'automazione verrà eseguita solo quando richiamata in modo esplicito mediante il package **APEX_AUTOMATION**.

[Visualizza documentazione](#)

Azioni avviate in

Selezionare se l'esecuzione delle azioni di automazione deve essere avviata in base a una **query** (le righe restituite possono quindi essere utilizzate dalle azioni) o se le azioni devono essere eseguite **sempre**.

[Visualizza documentazione](#)

Automazioni

✓

L'automazione è stata creata e si trova nello stato "Disabilitato". Creare le azioni da eseguire al momento dell'esecuzione dell'automazione.

✕

Automazione: Automazione 1

Annulla Elimina Salva modifiche Salva ed esegui

Mostra tutto

Nome Impostazioni Origine Azioni Esecuzione azione Esecuzione codice aggiuntiva Condizione lato server Avanzate

Nome

* Nome Automazione 1 ?

ID statico automazione-1 ?

Impostazioni

Tipo Su richiesta **Planificato** ?

* Espressione di pianificazione FREQ=HOURLY; INTERVAL=1; BYMINUTE=0

Stato pianificazione Attivo **Disabilitato** ?

Azioni avviate in Query Corpo funzione che restituisce un valore booleano Sempre ?

Origine

Origine dati Database locale Servizio SQL abilitato per REST Origine dati REST ?

Tipo di origine Tabella Query SQL Corpo funzione che restituisce SQL ?

Proprietario tabella/vista Schema di analisi ?

* Nome tabella/vista DIS_STRUTTURE_RICETTIVE (tabella) ?

Clausola Where

Clausola Order By

Includi colonna ROWID ?

Automazione: Automazione 1

Annulla Elimina Salva modifiche Salva ed esegui

Mostra tutto

Nome Impostazioni Origine Azioni Esecuzione azione Esecuzione codice aggiuntiva Condizione lato server Avanzate

Clausola Order By

Includi colonna ROWID ?

Azioni

Aggiungi azione

Q

Cerca: Tutte le colonne di testo

Vai

Modifica

Reimposta

✓		Nome	Sequenza di esecuzione	Tipo di azione	Posizione
✓		Nuova azione	10	Esegui codice	Database locale

1 righe selezionate Totale 1

Esecuzione azione

Esegui azioni quando **Righe restituite** Nessuna riga restituita ?

Colonna chiave primaria - Seleziona - ?

Commit **Una volta** Ogni riga ?

Numero massimo di righe da elaborare ?

Gestione errori di azione **Ignora** Interrompi automazione Disabilita automazione ?

Esecuzione codice aggiuntiva

Codice PL/SQL eseguibile ?

↶ ↷ 🔍 A= ✓

⚙️

Automazioni

APEX

App Builder

SQL Workshop

Team Development

Galleria

↑

Applicazione 766 \ Componenti condivisi \ Automazioni \ Automazione 1 \ Modifica azione

Modifica azione

Mostra tutto

Azione

Modifica azione

Codice

Gestione errori

Condizione lato server

Avanzate

Azione

* Nome

Nuova azione

* Tipo

Esegui codice

* Sequenza di esecuzione

10

Codice

Posizione

Database locale

Servizio SQL abilitato per REST

Lingua

PL/SQL

Codice

↶

↷

🔍

Aa

✓

3

end;

▼ Esempi

Esempio 1: inviare un messaggio di posta elettronica. In questo caso, si presume l'esistenza di un modello di posta elettronica denominato ORDER, con i segnaposto usati nel parametro p_placeholders.

```
begin
  apex_mail.send (
    p_template_static_id => 'ORDER',
    p_placeholders      => '{ "ORDER_NUMBER": ' || :ORDER_NUMBER ||
                          '"ORDER_DATE":' || to_char( :ORDER_DATE, 'YYYY-MM-DD' ) || ', '
                          '"ORDER_TOTAL":' || :ORDER_TOTAL ||
                          '}',
    p_to                => 'some_user@somewhere.com' );
end;
```

Esempio 2: inserire una riga in un'altra tabella. Se il valore :SAL è NULL, scrive un messaggio di avvertenza nel log di automazione.

```
begin
  insert into other_table( empno, ename, sal )
    values( :EMPNO, :ENAME, :SAL );
  if :SAL is null then
    apex_automation.log_warn( 'EMPNO ' || :EMPNO || ' has a NULL Salary');
  end if;
end;
```

Gestione errori

Messaggio di errore

Arresto esecuzione in caso di errore

☒

Condizione lato server

Tipo di condizione

- Seleziona tipo di condizione -

Automazioni

Azione

* Nome

Nuova azione

?

* Tipo

Invia messaggio di posta elettronica

?

* Sequenza di esecuzione

10

?

Invia impostazioni di posta elettronica

* Da

&APP_EMAIL.

?

* In

Cc

Ccn

Rispondi a

Modello di posta elettronica

- Nessun modello -

?

* Oggetto

* Testo non codificato corpo

HTML corpo

?

1

HTML corpo

?

1

SQL allegato

?

1

SQL allegato

Immettere un'istruzione SQL che restituisca uno o più BLOB allegati come file al messaggio di posta elettronica.

Se la colonna content_id contiene un valore nullo, i file associati saranno allegati standard. File come documenti o archivi zip vengono, in genere, allegati in questo modo.

Se il valore di content_id non è nullo, il file associato verrà contrassegnato come allegato in linea. Il valore della colonna content_id verrà utilizzato come CID (ID contenuto) dell'allegato. Questo valore CID può essere utilizzato nel codice HTML del messaggio di posta elettronica per incorporare le immagini direttamente nel contenuto del messaggio. Tenere presente che alcuni client di posta elettronica non possono visualizzare automaticamente gli allegati in linea. Inoltre, gli standard di posta elettronica prevedono vari requisiti per il formato degli identificativi di contenuto. Oracle APEX non garantisce la conformità a questi standard.

L'istruzione SQL deve avere il seguente formato:

```
select blob_column,
       file_name,
       mime_type,
       content_id
from table
```

Le colonne mime_type e content_id sono facoltative. Se è presente content_id, il file verrà allegato in linea e il valore di content_id verrà utilizzato per il CID dell'allegato.

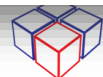
Se la colonna content_id contiene il valore 'my_inline_content', l'immagine può essere incorporata nel messaggio di posta elettronica utilizzando il seguente codice HTML:

```
<div>
  <h1>Contenuto incorporato</h1>
  <p>
    
  </p>
</div>
```

Visualizza documentazione

ORACLE

Partner

 **IEMA**
Software Architectures

capancioni.com

Automazioni

```
APEX_AUTOMATION.EXECUTE (
    p_application_id      IN NUMBER                DEFAULT {current application id},
    p_static_id           IN VARCHAR2,
    p_filters              IN apex_exec.t_filters   DEFAULT apex_exec.c_empty_filters,
    p_order_bys            IN apex_exec.t_order_bys DEFAULT apex_exec.c_empty_order_bys )
```

```
DECLARE
    l_filters apex_exec.t_filters;
BEGIN
    apex_session.create_session( 100, 1, 'ADMIN' );

    apex_exec.add_filter(
        p_filters      => l_filters,
        p_column_name  => 'DEPTNO',
        p_filter_type   => apex_exec.c_filter_eq,
        p_value         => 10 );

    apex_automation.execute(
        p_static_id     => 'my_emp_table_automation',
        p_filters        => l_filters );
END;
```

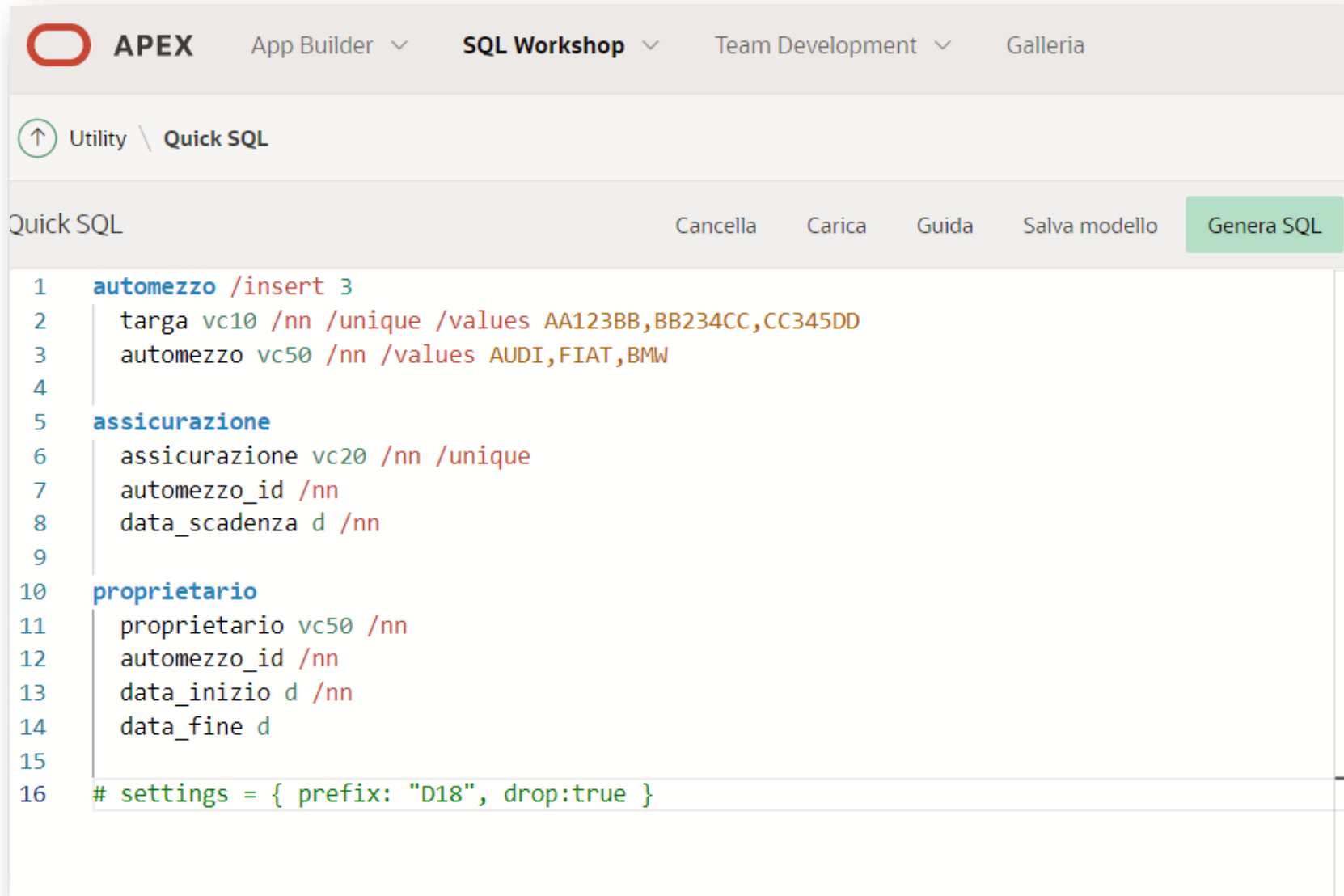
```
DECLARE
    l_context apex_exec.t_context;
BEGIN
    apex_session.create_session( 100, 1, 'ADMIN' );

    l_context := apex_exec.open_query_context(
        p_location  => apex_exec.c_location_local_db,
        p_sql_query => 'select * from emp_copy_table' );

    apex_automation.execute(
        p_static_id     => 'my_emp_table_automation',
        p_query_context  => l_context );
END;
```

D18 Automezzo – Dati Storizzati

D18 Automezzo – Dati Storizzati

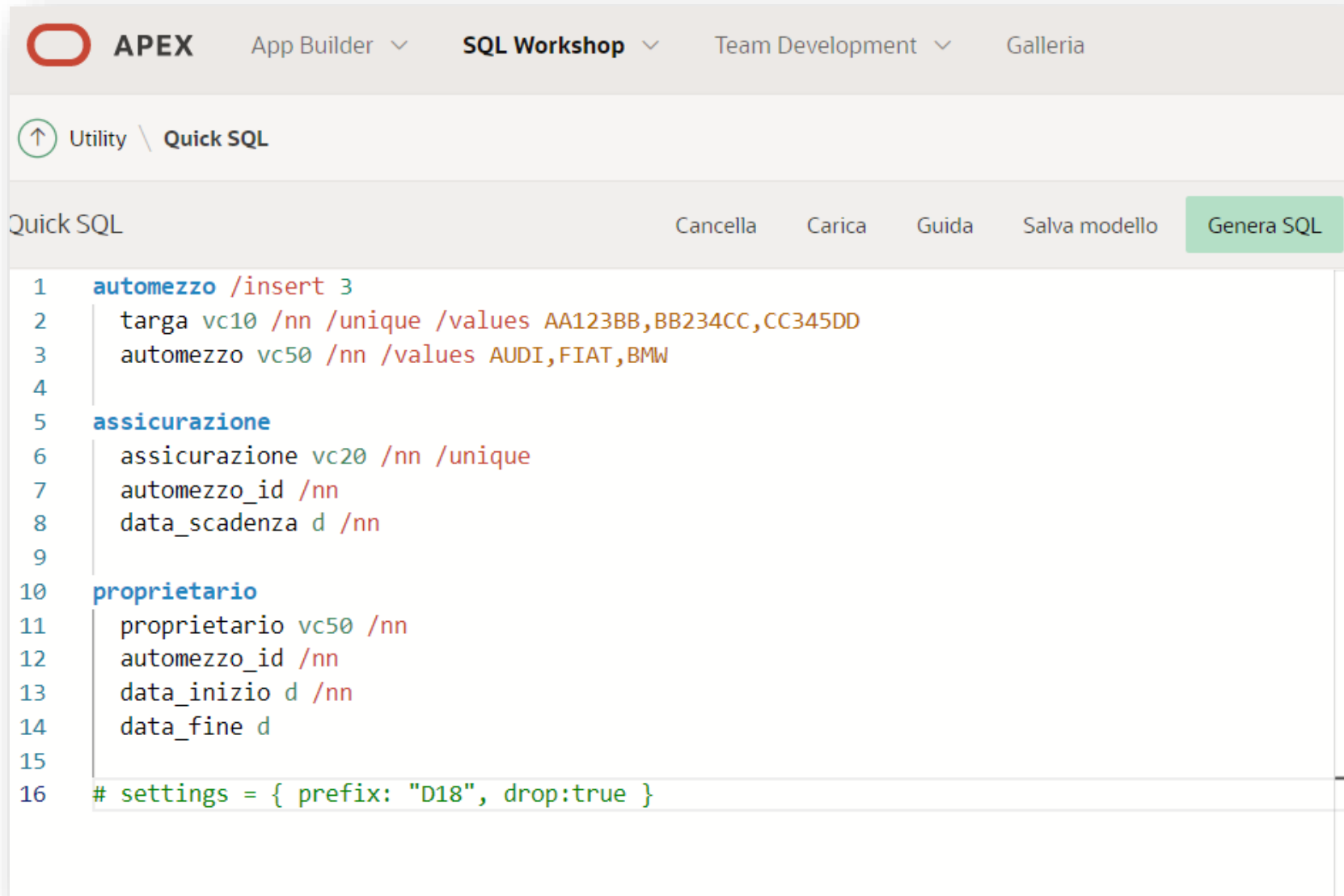


The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Galleria'. Below this, a breadcrumb trail shows 'Utility \ Quick SQL'. The main area is titled 'Quick SQL' and contains a text editor with the following SQL code:

```
1 automezzo /insert 3
2   targa vc10 /nn /unique /values AA123BB,BB234CC,CC345DD
3   automezzo vc50 /nn /values AUDI,FIAT,BMW
4
5 assicurazione
6   assicurazione vc20 /nn /unique
7   automezzo_id /nn
8   data_scadenza d /nn
9
10 proprietario
11   proprietario vc50 /nn
12   automezzo_id /nn
13   data_inizio d /nn
14   data_fine d
15
16 # settings = { prefix: "D18", drop:true }
```

At the top right of the editor, there are buttons for 'Cancella', 'Carica', 'Guida', 'Salva modello', and a green 'Genera SQL' button.

D18 Automezzo – Dati Storizzati



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Galleria'. Below this is a breadcrumb trail 'Utility \ Quick SQL'. The main area is titled 'Quick SQL' and contains a text editor with the following SQL code:

```
1 automezzo /insert 3
2   targa vc10 /nn /unique /values AA123BB,BB234CC,CC345DD
3   automezzo vc50 /nn /values AUDI,FIAT,BMW
4
5 assicurazione
6   assicurazione vc20 /nn /unique
7   automezzo_id /nn
8   data_scadenza d /nn
9
10 proprietario
11   proprietario vc50 /nn
12   automezzo_id /nn
13   data_inizio d /nn
14   data_fine d
15
16 # settings = { prefix: "D18", drop:true }
```

At the top right of the editor area are buttons for 'Cancella', 'Carica', 'Guida', 'Salva modello', and 'Genera SQL'.

D18 Automezzo – Dati Storicizzati

```
create unique index d18_assicurazione_uq1 on d18_assicurazione  
(automezzo_id,data_scadenza);
```

```
create unique index d18_proprietario_uq1 on d18_proprietario  
(automezzo_id,data_fine);
```

D18 Automezzo – Dati Storicizzati

```
select m.targa,  
       a.assicurazione,  
       a.data_scadenza,  
       lag(a.data_scadenza) over (partition by a.automezzo_id  
                                  order by a.data_scadenza) data_scadenza_prec  
from d18_assicurazione a  
join d18_automezzo m      on a.automezzo_id = m.id
```

D18 Automezzo – Dati Storici

```
with assicurazione as (  
select m.targa,  
       a.assicurazione,  
       a.data_scadenza,  
       lag(a.data_scadenza) over (partition by a.automezzo_id  
                                  order by a.data_scadenza) data_scadenza_prec  
from d18_assicurazione a  
join d18_automezzo m      on a.automezzo_id = m.id)  
select *  
from assicurazione  
where date '2022-01-01' between nvl(data_scadenza_prec, date '1900-01-01')  
and data_scadenza
```

D18 Automezzo – Dati Storici

```
with assicurazione as (  
select m.targa,  
       a.assicurazione,  
       a.data_scadenza,  
       lag(a.data_scadenza) over (partition by a.automezzo_id  
                                   order by a.data_scadenza) data_scadenza_prec,  
       lead(a.data_scadenza) over (partition by a.automezzo_id  
                                   order by a.data_scadenza) data_scadenza_succ  
from d18_assicurazione a  
join d18_automezzo m      on a.automezzo_id = m.id)  
select *  
from assicurazione  
where data_scadenza_succ is null
```