

ORACLE

Corso APEX ODCEC Milano: Edizione 2024

Lezione 24 - 8 Luglio 2024

Ing. Roberto Capancioni



Chi Sono

Ing. Roberto Capancioni
—Trainer Oracle Academy

capancioni.com



Email: roberto.capancioni@oracle.com
Linkedin: <https://www.linkedin.com/in/robertocapancioni>



Tabelle Blockchain

Tabelle Blockchain

Quick SQL

Guida Impostazioni

SQL Diagramma

```
1 persona
2   nome vc30 /nn
3   data d /nn
4
5 #prefix:d24
6
```

```
1 -- create tables
2
3 create table d24_persona (
4     id      number generated by default on null as identity
5     |      | constraint d24_persona_id_pk primary key,
6     nome    varchar2(30 char) not null,
7     data    date not null
8 );
9
10
```

```
create table d24_persona (
    id          number generated by default on null as identity
               constraint d24_persona_id_pk primary key,
    nome        varchar2(30 char) not null,
    data        date not null
);
```

```
create blockchain table d24_persona (
    id          number generated by default on null as identity
               constraint d24_persona_id_pk primary key,
    nome        varchar2(30 char) not null,
    data        date not null
)
no drop until 0 days idle
no delete until 31 days after insert
hashing using "sha2 512" version v1;
```



Tabelle Blockchain

```
create blockchain table d24_persona (  
    id          number generated by default on null as identity  
                constraint d24_persona_id_pk primary key,  
    nome        varchar2(30 char) not null,  
    data        date not null  
)  
no drop until 0 days idle  
no delete until 31 days after insert  
hashing using "sha2_512" version v1;
```

BLOCKCHAIN

Oracle Blockchain Table

```
CREATE BLOCKCHAIN TABLE  
trade_ledger (...);
```

TRADE LEDGER

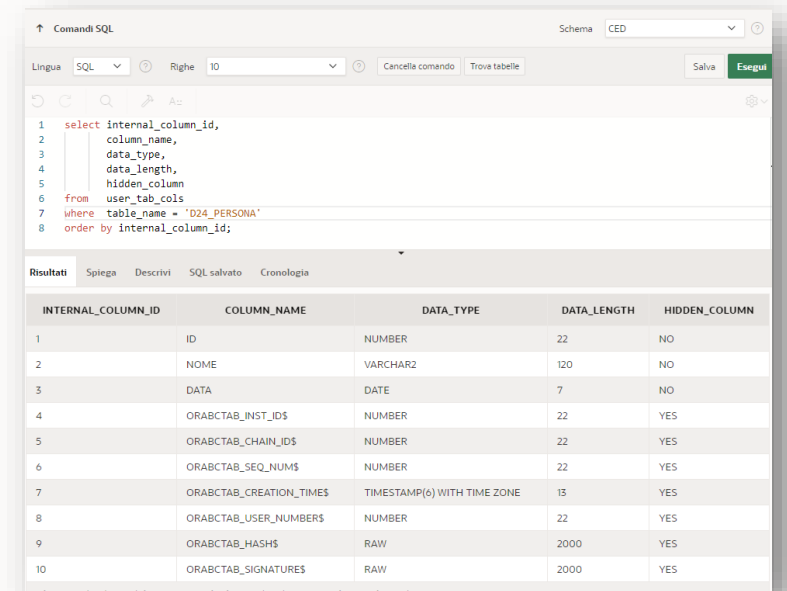
ID	User	Value	Created	CryptoDigest
1	Tom	500	1-Feb	ADSJS
2	Carol	176	8-Mar	%10S
3	Wang	500	3-Aug	SH31
4	Eve	25	14-Oct	LRO\$

Tabelle Blockchain

```
insert into d24_persona (nome,data) values ('Roberto',sysdate);  
insert into d24_persona (nome,data) values ('Maria',sysdate);
```

```
select internal_column_id,  
       column_name,  
       data_type,  
       data_length,  
       hidden_column  
from user_tab_cols  
where table_name = 'D24_PERSONA'  
order by internal_column_id;
```

Query per elencare tutte le
colonne di una tabella
(anche quelle nascoste)



The screenshot shows the SQL Developer interface. The top pane contains the SQL query: `select internal_column_id, column_name, data_type, data_length, hidden_column from user_tab_cols where table_name = 'D24_PERSONA' order by internal_column_id;`. The bottom pane displays the results in a table with 5 columns: INTERNAL_COLUMN_ID, COLUMN_NAME, DATA_TYPE, DATA_LENGTH, and HIDDEN_COLUMN. There are 10 rows of data.

INTERNAL_COLUMN_ID	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	HIDDEN_COLUMN
1	ID	NUMBER	22	NO
2	NOME	VARCHAR2	120	NO
3	DATA	DATE	7	NO
4	ORABCTAB_INST_ID\$	NUMBER	22	YES
5	ORABCTAB_CHAIN_ID\$	NUMBER	22	YES
6	ORABCTAB_SEQ_NUM\$	NUMBER	22	YES
7	ORABCTAB_CREATION_TIME\$	TIMESTAMP(6) WITH TIME ZONE	13	YES
8	ORABCTAB_USER_NUMBERS\$	NUMBER	22	YES
9	ORABCTAB_HASH\$	RAW	2000	YES
10	ORABCTAB_SIGNATURE\$	RAW	2000	YES

```
select p.*, orabctab_hash$ from d24_persona p;
```

ID	NOME	DATA	ORABCTAB_HASH\$
1	Roberto	08/07/2024	64A3DAA520EB6D4322CAFO69765EB598D37DCFDCEE9B02C983C5220461B68BDE4751EF9F5ACFDA946C892E694F37ID445ADFA9AB23928A24BAD1A9F2B530B903
2	Maria	08/07/2024	7E99E29354D4EDAC2547091676094D623106CA6561B1995BB654000E80987AD287EE8057AAB72BC7048CC915E947D3A8A91A93BA031EB1308735DB66BEA30394

Tabelle Blockchain – NO DROP UNTIL

```
create blockchain table d24_persona (  
    id          number generated by default on null as identity  
                constraint d24_persona_id_pk primary key,  
    nome        varchar2(30 char) not null,  
    data        date not null  
)  
no drop until 0 days idle  
no delete until 31 days after insert  
hashing using "sha2_512" version v1;
```

NO DROP UNTIL <num> DAYS IDLE

Tabelle Blockchain – NO DROP UNTIL

```
create blockchain table d24_persona (  
    id          number generated by default on null as identity  
                constraint d24_persona_id_pk primary key,  
    nome        varchar2(30 char) not null,  
    data        date not null  
)  
no drop until 0 days idle  
no delete until 31 days after insert  
hashing using "sha2_512" version v1;
```

```
alter table d24_persona no drop until 2 days idle;  
alter table d24_persona no drop until 1 days idle;
```

```
drop table d24_persona;
```

Posso incrementare i giorni ma non
posso diminuirli

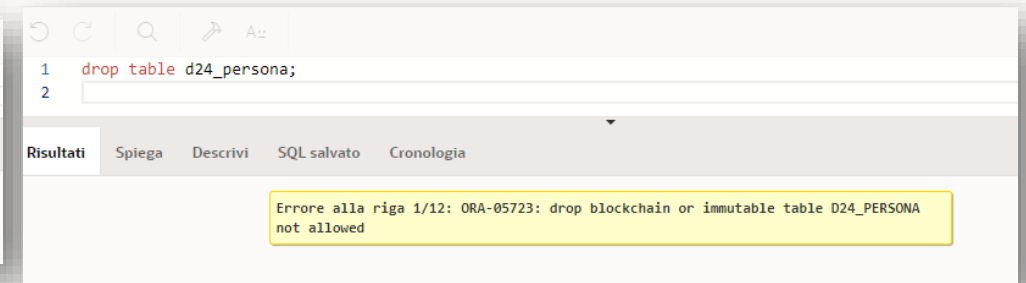
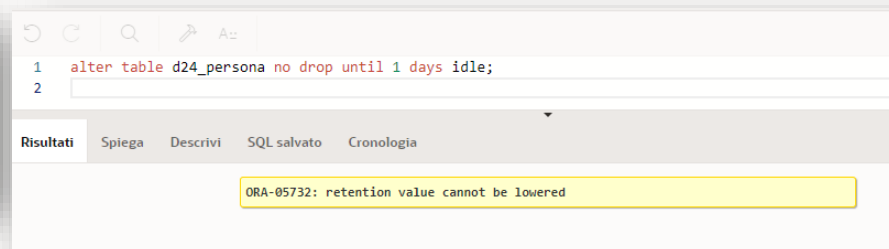
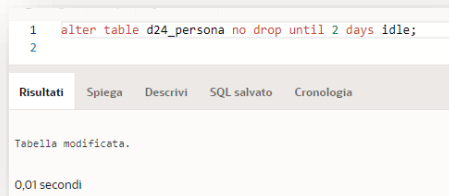


Tabelle Blockchain – NO DELETE UNTIL

```
create blockchain table d24_persona (  
    id          number generated by default on null as identity  
                constraint d24_persona_id_pk primary key,  
    nome        varchar2(30 char) not null,  
    data        date not null  
)  
no drop until 0 days idle  
no delete until 31 days after insert  
hashing using "sha2_512" version v1;
```

```
alter table d24_persona no delete until 32 days after insert;  
alter table d24_persona no delete until 30 days after insert;
```

```
delete from d24_persona;
```

Posso incrementare i giorni ma non
posso diminuirli

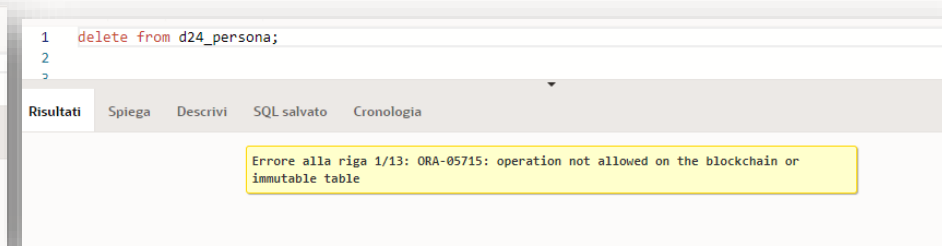
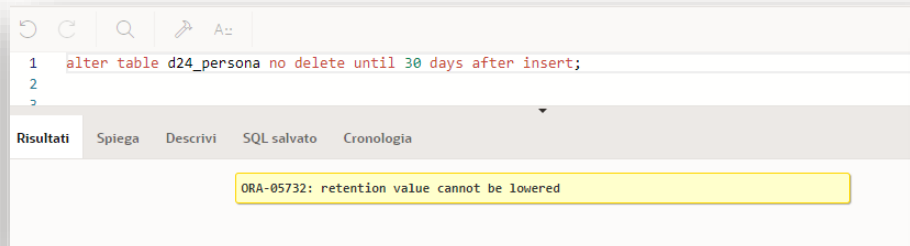
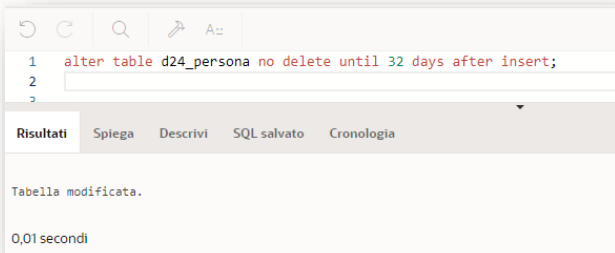


Tabelle Blockchain – Cancellare righe “scadute”

```
declare
  l_rows number;
begin
  dbms_blockchain_table.delete_expired_rows(
    schema_name      => 'X250',
    table_name       => 'D24_PERSONA',
    before_timestamp  => null,
    number_of_rows_deleted => l_rows);

  dbms_output.put_line('Riche Cancellate=' || l_rows);
end;
/
```

Tutte le scadute

```
declare
  l_rows number;
begin
  dbms_blockchain_table.delete_expired_rows(
    schema_name      => 'X250',
    table_name       => 'D24_PERSONA',
    before_timestamp  => systimestamp - 60,
    number_of_rows_deleted => l_rows);

  dbms_output.put_line('Riche Cancellate=' || l_rows);
end;
/
```

Scadute ma con una
condizione

Tabelle Blockchain – Verifica righe

```
declare
    l_rows      number;
    l_verified   number;
begin
    select count(*)
    into   l_rows
    from   d24_persona;

    dbms_blockchain_table.verify_rows(
        schema_name      => 'X250',
        table_name        => 'D24_PERSONA',
        number_of_rows_verified => l_verified);

    dbms_output.put_line('Righe=' || l_rows || ' Righe
Verificate=' || l_verified);
end;
/
```

Se uno avesse accesso fisicamente al server
potrebbe modificare dati esternamente al
database agendo su file fisici

Ma noi possiamo sempre verificare l'integrità
rispetto alla prima riga

```
63 declare
64     l_rows      number;
65     l_verified   number;
66 begin
67     select count(*)
68     into   l_rows
69     from   d24_persona;
70
71     dbms_blockchain_table.verify_rows(
72         schema_name      => 'CED',
73         table_name        => 'D24_PERSONA',
74         number_of_rows_verified => l_verified);
75
76     dbms_output.put_line('Righe=' || l_rows || ' Righe Verificate=' || l_verified);
77 end;
78 /
```

Risultati Spiega Descrivi SQL salvato Cronologia

Righe=2 Righe Verificate=2

Istruzione elaborata.

0,02 secondi



Tabelle Blockchain – Firma Righe

E se compromettessero anche la prima riga?

E' possibile usare dei meccanismi di firma digitale tramite certificati

Manage Certificates

The DBMS_USER_CERTS package allows us to manage certificates for use with row signing.

We generated a new self-signed certificate using the following openssl command on a Linux server.

```
mkdir /home/oracle/my_wallet

openssl req \
  -newkey rsa:2048 -nodes -sha512 \
  -x509 -days 3650 \
  -outform der \
  -keyout /home/oracle/my_wallet/my-bct-test-key.der \
  -out /home/oracle/my_wallet/my-bct-test-cert.der \
  -subj "/C=GB/ST=West Midlands/L=Birmingham/O=Example Company/OU=Devs/CN=Tim Hall/emailAddress=me@example.com"
```

We create an oracle directory object pointing to the location holding the certificate.

APEX_STRING.SPLIT()

—

D24 - Template Directives

Template Directives

- Report Classico
 - Report Interattivo
 - Griglia Interattiva
 - Schede
 - Email templates
 - Mappe
 - Template Componets
- **Direttiva IF**
 - Mostra o meno il testo in base al valore di un elemento o colonna
 - **Direttiva CASE WHEN**
 - Mostra o meno il testo in base a più condizioni sul valore di un elemento o colonna.
 - **Direttiva LOOP**
 - Usa un loop per ripetere il testo per ogni element o Colonna di tipo multivalore (con un separatore di carattere)
 - **Direttiva WITH**
 - Usa template predefiniti nelle espressioni HTML

Template Directives

`[!]` E' una negazione

```
{if [!]NAME/}  
    testo1  
{elseif [!]NAME2/}  
    testo2  
{else/  
    altro testo  
{endif/}
```

```
{if DESCRIPTION/}  
    &DESCRIPTION.  
{else/  
    Nessuna Descrizione  
{endif/}
```

&**DESCRIPTION**. o #**DESCRIPTION**#
In base all'area

Template Directives

```
{case NAME/}  
{when valore1/  
    testo1  
{when valore2/  
    testo2  
{otherwise/  
    altro testo  
{endcase/}
```

```
{case JOB/  
{when SALESMAN/  
    &SAL. (&COMM.)  
{when PRESIDENT/  
    --  
{otherwise/  
    &SAL.  
{endcase/}
```

Template Directives

```
{loop ["SEP"] NAME/}  
    testo  
{endloop/}
```

```
<ul>  
    {loop ", " TAGS/}  
        <li>&APEX$I. - &APEX$ITEM.</li>  
    {endloop/}  
</ul>
```

- **APEX\$ITEM**
 - Valore dell'elemento corrente nella lista
- **APEX\$I**
 - Posizione dell'elemento corrente nella lista

Template Directives

```
{with/}  
    TYPE:=IMAGE  
    IMAGE:=#PROFILE_IMAGE#  
    IMAGE_ALT:=Profile Image  
    SIZE:=t-Avatar--lg  
    SHAPE:=t-Avatar--circle  
    CSS_CLASSES:=u-color-29  
{apply THEME$AVATAR/}
```

APEX_STRING.SPLIT

APEX_STRING è un package di APEX che contiene diverse procedure e funzioni per gestire stringhe

A noi interessa la funzione SPLIT



The screenshot shows the Oracle APEX interface. At the top, there's a toolbar with icons for undo, redo, search, and other functions. Below the toolbar, a SQL query is entered in a text area:

```
1 select /*+ cardinality (s, 2) */ s.*
2 from apex_string.split('452:330:164:163:380',':') s
3
```

Below the query, there's a tabbed interface with 'Risultati' (Results) selected. The results are displayed in a table with one column, 'COLUMN_VALUE'.

COLUMN_VALUE
452
330
164
163
380

At the bottom of the results table, it says '5 righe restituite in 0,01 secondi' (5 rows returned in 0.01 seconds) and a 'Scarica' (Download) button.

```
select column_value from apex_string.split('452:330:164:163:380',':');
```

Versione ottimizzata con performance maggiori

```
select /*+ cardinality (s, 20) */ column_value
from apex_string.split('452:330:164:163:380',':') s;
```