

Oracle Password Verify Function

Olá, neste artigo apresento a sintaxe necessária para criar a função de verificação de complexibilidade de senhas no banco de dados Oracle.

Version 12C ONLY with COMPATIBLE < 12.0.0.0:

```
CREATE OR REPLACE FUNCTION "SYS"."VERIFY_FUNCTION"
(username varchar2,
 password varchar2,
 old_password varchar2)
RETURN boolean IS
  differ integer;
  pw_lower varchar2(256);
  db_name varchar2(40);
  i integer;
  simple_password varchar2(10);
  reverse_user varchar2(32);
BEGIN
  IF NOT ora_complexity_check(password, chars => 14, letter => 1, digit =>
1) THEN
    RETURN(FALSE);
  END IF;

  -- Check if the password contains the username
  pw_lower := NLS_LOWER(password);
  IF instr(pw_lower, NLS_LOWER(username)) > 0 THEN
    raise_application_error(-20002, 'Password contains the username');
  END IF;

  -- Check if the password contains the username reversed
  reverse_user := '';
  FOR i in REVERSE 1..length(username) LOOP
    reverse_user := reverse_user || substr(username, i, 1);
  END LOOP;
  IF instr(pw_lower, NLS_LOWER(reverse_user)) > 0 THEN
    raise_application_error(-20003, 'Password contains the username ' ||
'reversed');
  END IF;

  -- Check if the password contains the server name
  select name into db_name from sys.v$database;
  IF instr(pw_lower, NLS_LOWER(db_name)) > 0 THEN
    raise_application_error(-20004, 'Password contains the server name');
  END IF;

  -- Check if the password contains 'oracle'
  IF instr(pw_lower, 'oracle') > 0 THEN
    raise_application_error(-20006, 'Password too simple');
  END IF;
```

```

-- Check if the password is too simple. A dictionary of words may be
-- maintained and a check may be made so as not to allow the words
-- that are too simple for the password.
IF pw_lower IN ('welcome1', 'database1', 'account1', 'user1234',
               'password1', 'oracle123', 'computer1',
               'abcdefg1', 'change_on_install') THEN
    raise_application_error(-20006, 'Password too simple');
END IF;

-- Check if the password differs from the previous password by at least
-- 3 characters
IF old_password IS NOT NULL THEN
    differ := ora_string_distance(old_password, password);
    IF differ < 3 THEN
        raise_application_error(-20010, 'Password should differ from the '
                                || 'old password by at least 3 characters');
    END IF;
END IF ;

RETURN(TRUE);
END;
/

```

- Lembrando que é possível adaptar a função conforme a necessidade e políticas de segurança.

Version 12C ONLY with COMPATIBLE > 11.2.0.4:

```

CREATE OR REPLACE NONEDITIONABLE FUNCTION SYS.VERIFY_FUNCTION
(username varchar2,
 password varchar2,
 old_password varchar2)
RETURN boolean IS
    differ integer;
BEGIN
    -- Check if the password is same as the username
    IF NLS_LOWER(password) = NLS_LOWER(username) THEN
        raise_application_error(-20001, 'Password same as or similar to user');
    END IF;

    -- Check if the password contains at least four characters, including
    -- one letter, one digit and one punctuation mark.
    IF NOT ora_complexity_check(password, chars => 14, letter => 1, digit =>
1) T
HEN
        RETURN(FALSE);
    END IF;

    -- Check if the password is too simple. A dictionary of words may be
    -- maintained and a check may be made so as not to allow the words
    -- that are too simple for the password.
    IF NLS_LOWER(password) IN ('welcome', 'database', 'account', 'user',

```

```

        'password', 'oracle', 'computer', 'abcd',
        'welcome1', 'database1', 'account1',
        'user1234', 'password1', 'oracle123',
        'computer1', 'abcdefg1', 'change_on_install')
THEN

    raise_application_error(-20002, 'Password too simple');
END IF;

-- Check if the password differs from the previous password by at least
-- 3 letters
IF old_password IS NOT NULL THEN
    differ := ora_string_distance(old_password, password);
    IF differ < 3 THEN
        raise_application_error(-20004, 'Password should differ by at' ||
            'least 3 characters');
    END IF;
END IF;
RETURN(TRUE);
END;
/

```

- Lembrando que é possível adaptar a função conforme a necessidade e políticas de segurança.

CDB LEVEL ==> 19C ONLY:

```

CREATE OR REPLACE NONEDITIONABLE FUNCTION SYS.VERIFY_FUNCTION
(username varchar2,
 password varchar2,
 old_password varchar2)
RETURN boolean IS
    differ integer;
    canon_username dbms_id := username;
BEGIN
    -- Bug 22369990: Dbms_Utility may not be available at this point, so
switch
    -- to dynamic SQL to execute canonicalize procedure.
    IF (substr(username,1,1) = '') THEN
        execute immediate 'begin dbms_utility.canonicalize(:p1, :p2, 128);
end; '
        using IN username, OUT canon_username;
    END IF;
    -- Check if the password is same as the username
    IF NLS_LOWER(password) = NLS_LOWER(canon_username) THEN
        raise_application_error(-20001, 'Password same as or similar to user');
    END IF;

    -- Check if the password contains at least four characters, including
    -- one letter, one digit and one punctuation mark.
    IF NOT ora_complexity_check(password, chars => 14, letter => 1, digit =>
1) THEN

```

```

    RETURN(FALSE);
END IF;

-- Check if the password is too simple. A dictionary of words may be
-- maintained and a check may be made so as not to allow the words
-- that are too simple for the password.
IF NLS_LOWER(password) IN ('welcome', 'database', 'account', 'user',
                          'password', 'oracle', 'computer', 'abcd') THEN
    raise_application_error(-20002, 'Password too simple');
END IF;

-- Check if the password differs from the previous password by at least
-- 3 letters
IF old_password IS NOT NULL THEN
    differ := ora_string_distance(old_password, password);
    IF differ < 3 THEN
        raise_application_error(-20004, 'Password should differ by at' ||
                                'least 3 characters');
    END IF;
END IF;

RETURN(TRUE);
END;
/

```

- Lembrando que é possível adaptar a função conforme a necessidade e políticas de segurança.

Abaixo um exemplo básico de um profile utilizando a função de verificação de complexidade de senhas **NON-PDB AND PDB LEVEL**:

```

CREATE PROFILE END_USER
LIMIT
    COMPOSITE_LIMIT DEFAULT
    SESSIONS_PER_USER DEFAULT
    CPU_PER_SESSION DEFAULT
    CPU_PER_CALL DEFAULT
    LOGICAL_READS_PER_SESSION DEFAULT
    LOGICAL_READS_PER_CALL DEFAULT
    IDLE_TIME DEFAULT
    CONNECT_TIME DEFAULT
    PRIVATE_SGA DEFAULT
    FAILED_LOGIN_ATTEMPTS 5
    PASSWORD_LIFE_TIME 15552000/86400
    PASSWORD_REUSE_TIME DEFAULT
    PASSWORD_REUSE_MAX 12
    PASSWORD_VERIFY_FUNCTION "VERIFY_FUNCTION"
    PASSWORD_LOCK_TIME 900/86400
    PASSWORD_GRACE_TIME DEFAULT
/

```

- Até a próxima.

Oracle Password Verify Function

— Autor: *Leonardo Lopes* 28/11/2024 10H:21

From:

<https://blog.dbahero.com/> -

Permanent link:

<https://blog.dbahero.com/doku.php?id=oraclepwdverifyfunction>

Last update: **28/11/2024 10H:21**

