

# i2i Academy

## Training Document

<b>Topic</b>	SQL Language Fundamentals I
<b>Document Name</b>	PLSQL01-EX-08

Document Difficulty Level			
Beginner	Junior	Senior	Expert
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Copyright of  i2i Systems Turkey 2013

The copyright in this work is vested in i2i Systems Turkey and the information contained herein is confidential. This work (either in whole or in part) must not be modified, reproduced, disclosed or disseminated to others or used for purposes other than that for which it is supplied, without the prior written permission of i2i Systems Turkey. If this work or any part hereof is furnished to a third party by virtue of a contract with that party, use of this work by such party shall be governed by the express contractual terms between the i2i Systems Turkey which is a party to that contract and the said party.

---

## Document History

Date	Author	Ver	Comments
<b>20.09.2013</b>	Onur Bayram	1.0	Initial Draft
<b>03.10.2013</b>	Erman İbrişim	1.1	Sample Outputs and Rules Added

## SQL Language Fundamentals I

### Exercise PLSQL01-EX-08:

**Definiton :** Write a PL/SQL program that prints out string in xml format.

Input String : “acbbcadefghkkgfed”

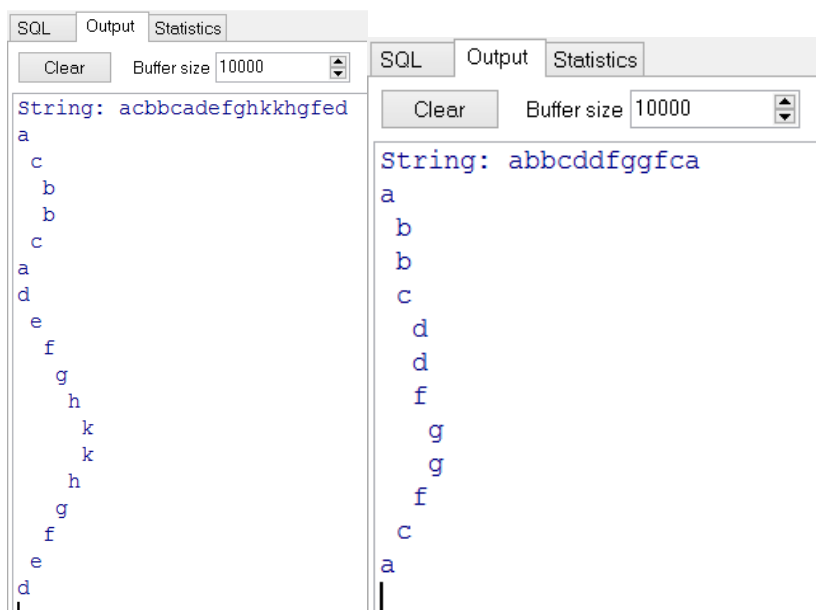
Input String: “abbcdffggfca”

**Rules:** -Maximum string length is 30.

-No duplicate characters in string like ‘abbcdffaafca’ or ‘defgddgfed’.

-String can’t have space character or any numeric characters.

**Sample Output :**



The image shows two side-by-side screenshots of the SQL Developer interface, specifically the Output window. Both windows have tabs for SQL, Output, and Statistics, and a 'Clear' button and a 'Buffer size' dropdown set to 10000.

The left screenshot shows the output for the input string "acbbcadefghkkgfed". The output is an XML string where each character is enclosed in a separate element, and the elements are nested to form a tree structure. The output is: `String: acbbcadefghkkgfed`  
`a`  
`c`  
`b`  
`b`  
`c`  
`a`  
`d`  
`e`  
`f`  
`g`  
`h`  
`k`  
`k`  
`h`  
`g`  
`f`  
`e`  
`d`

The right screenshot shows the output for the input string "abbcdffggfca". The output is an XML string where each character is enclosed in a separate element, and the elements are nested to form a tree structure. The output is: `String: abbcdffggfca`  
`a`  
`b`  
`b`  
`c`  
`d`  
`d`  
`f`  
`g`  
`g`  
`f`  
`c`  
`a`

**Objectives :** To learn SQL string functions and conditional clauses(IF THEN ELSE), loops(FOR, WHILE)

## Solution of PLSQL01-EX-08:

Kullanılan kod:

```
DECLARE input_string VARCHAR2(30) := 'acbbcadefghkkgfed'; -- örnek string current_char CHAR(1);
seen_chars VARCHAR2(1000) := ''; i INTEGER; BEGIN -- 1. Giriş string'ini yazdır
DBMS_OUTPUT.PUT_LINE('String: ' || input_string);

-- 2. Kurallar:
IF LENGTH(input_string) > 30 THEN
    RAISE_APPLICATION_ERROR(-20001, 'String çok uzun.');
```

END IF;

```
FOR i IN 1 .. LENGTH(input_string) LOOP
    current_char := SUBSTR(input_string, i, 1);

    -- Boşluk kontrolü
    IF current_char = ' ' THEN
        RAISE_APPLICATION_ERROR(-20002, 'String boşluk içeremez.');
```

END IF;

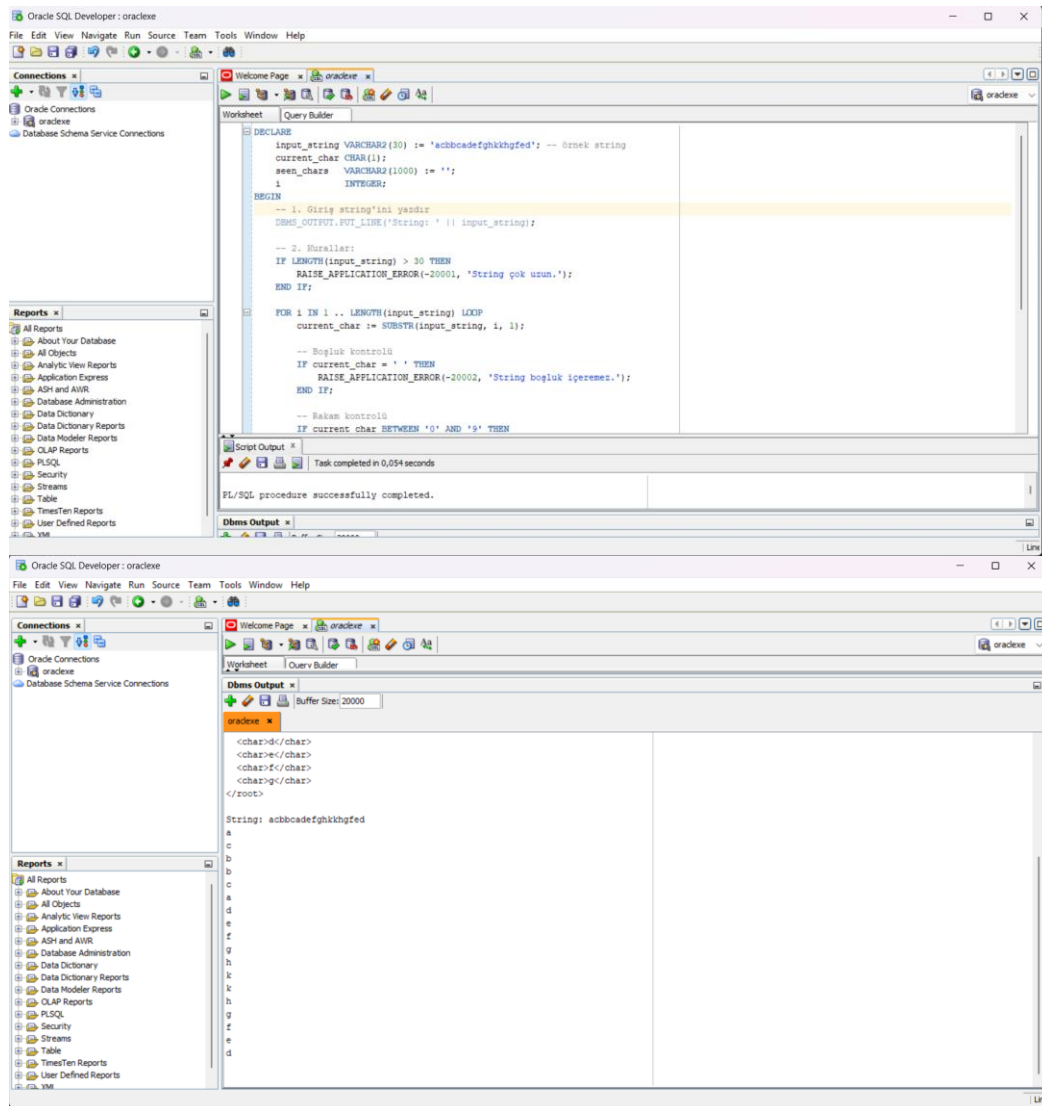
```
    -- Rakam kontrolü
    IF current_char BETWEEN '0' AND '9' THEN
        RAISE_APPLICATION_ERROR(-20003, 'String rakam içeremez.');
```

END IF;

```
    -- Tekrar eden karakter kontrolü
    IF INSTR(seen_chars, current_char) > 0 THEN
        -- Tekrar var, sadece yazdır, ama hata üretme
        NULL;
    ELSE
        seen_chars := seen_chars || current_char;
    END IF;

    -- Karakteri yazdır
    DBMS_OUTPUT.PUT_LINE(current_char);
END LOOP;
```

END;



The image displays two screenshots of the Oracle SQL Developer interface. The top screenshot shows a PL/SQL procedure being executed. The code defines a variable `input\_string` with the value 'acbbcadeqghkghgfed', initializes a cursor, and uses a loop to process the string character by character, raising an error if a character is not between 'a' and 'z'. The bottom screenshot shows the 'DBMS Output' window, which displays the execution results. The output shows the string 'acbbcadeqghkghgfed' and a list of characters 'a', 'c', 'b', 'd', 'e', 'g', 'h', 'k', 'f', 'e', 'd'.

**Top Screenshot: PL/SQL Code**

```

DECLARE
input_string VARCHAR2(30) := 'acbbcadeqghkghgfed'; -- Örnek string
current_char CHAR(1);
seen_chars VARCHAR2(1000) := '';
1
INTEGER;
BEGIN
-- 1. Giriş string'ini yazar
DBMS_OUTPUT.PUT_LINE('String: ' || input_string);

-- 2. Kontrol:
IF LENGTH(input_string) > 30 THEN
RAISE_APPLICATION_ERROR(-20001, 'String çok uzun.');
```

**Bottom Screenshot: DBMS Output**

```

<char><d/>char>
<char><c/>char>
<char><b/>char>
<char><d/>char>
<char><e/>char>
</root>

String: acbbcadeqghkghgfed
a
c
b
d
e
g
h
k
f
e
d
```