

**AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Course No: CSE4126

Course Title: Distributed Database Systems Lab

Fall 2019 | Lab Final Quiz | Marks 30 | Time: 70 Minutes

1. Why should we use **Function** when the **Procedure** can output value? Explain with example. 5
2. Read the following points carefully and answer accordingly. ***All of the following tasks should be done in a single sql file.***

Create a sql file and name it "Trigger\_160204XXX"

Assume that, we have a table named "STUDENTS" that has two attributes – *STUDENT\_NAME* which is of type varchar2 and *CGPA* of type number. Also assume that, there are already two rows inserted in "STUDENTS" table as follows –

STUDENT_NAME	CGPA
Walter White	3.71
Jesse Pinkman	3.20

- a. Create five triggers (FUN1, FUN2, ..., FUN5) so that when we run the following queries they will work as expected: 5

**INSERT INTO STUDENTS VALUES ('Gus Fring', 3.50);**

(This will run triggers FUN1, FUN2, FUN3 but FUN2 will be triggered before insertion.)

**DELETE FROM STUDENTS WHERE CGPA < 3.65;**

(This will run FUN3 but will not run FUN1 and FUN2)

**UPDATE STUDENTS SET CGPA = CGPA + 0.01 WHERE STUDENT\_NAME LIKE '%Fring%';**

(This will run FUN4, but will not run FUN5)

**UPDATE STUDENTS SET STUDENT\_NAME = 'Heisenberg' WHERE STUDENT\_NAME = 'Walter White';**

(This will not run any trigger)

**UPDATE STUDENTS SET CGPA = CGPA + 0.01;**

(This will run both FUN4 and FUN5)

*Note that, inside triggers you have to print the trigger name. For example – when the trigger FUN1 is executed it will print "FUN1", for FUN2 trigger - print "FUN2" and so on.*

- b. Create a PL/SQL trigger named '**INVALID\_NAME**' which will ensure that when a new row is inserted on the STUDENTS table, the name of a student contains only alphabetic characters. If the name is valid, then insertion is allowed otherwise, throw an exception from the trigger that would halt the insertion operation. 5
3.
  - a. Suppose, we want to execute **A.sql** file automatically from **B.sql** file. When we execute **B.sql**, **A.sql** will automatically execute. How can we achieve this? Explain with example. 2
  - b. Write down the purposes of **:OLD** and **:NEW** for a ROW LEVEL trigger on INSERT, UPDATE and DELETE operation. 2
  - c. Can we use array in PL/SQL? Give an example. 1
4. *TO\_NUMBER built in function can convert a VARCHAR2 value to a NUMERIC value in Oracle. When it cannot convert the varchar2 value, it throws an exception.* 5

Write a PL/SQL function **CONVERT\_NUMBER** that receives a VARCHAR2 value. The function will check whether the input can be converted to a valid number. If possible then CONVERT\_NUMBER will return 'YES', otherwise will return 'NO'. CONVERT\_NUMBER function should be declared inside a package named **PK\_CONVERSION**. All the codes will be in a single sql file named "**Task4.sql**"

*Note: You may use SUBSTR() built-in function.*

5. Write a PL/SQL code in a single sql file named "**Task5.sql**" to implement the following – 5
  - a. Prompt to user: "Enter last 3 digits of your ID = "
  - b. Nested Anonymous Blocks
  - c. Take Input in the outer block
  - d. Check Even/Odd in the inner block
  - e. Print "Even"/"Odd" in the outer block

*Note: It is possible to take input inside PL/SQL Begin-End block. This is not a valid/Invalid type question answer. If you do not know, search on google, you have the internet.*

### **Instructions:**

1. You will submit all the files in ZIP format.
2. Name the ZIP file with your ID. Example – 160204XXX.zip
3. Question 1 and 3 will be handwritten.
4. In your zip file – There will be 3 sql files (for question – 2, 4, 5) and 1 pdf file (for question 1, 3)