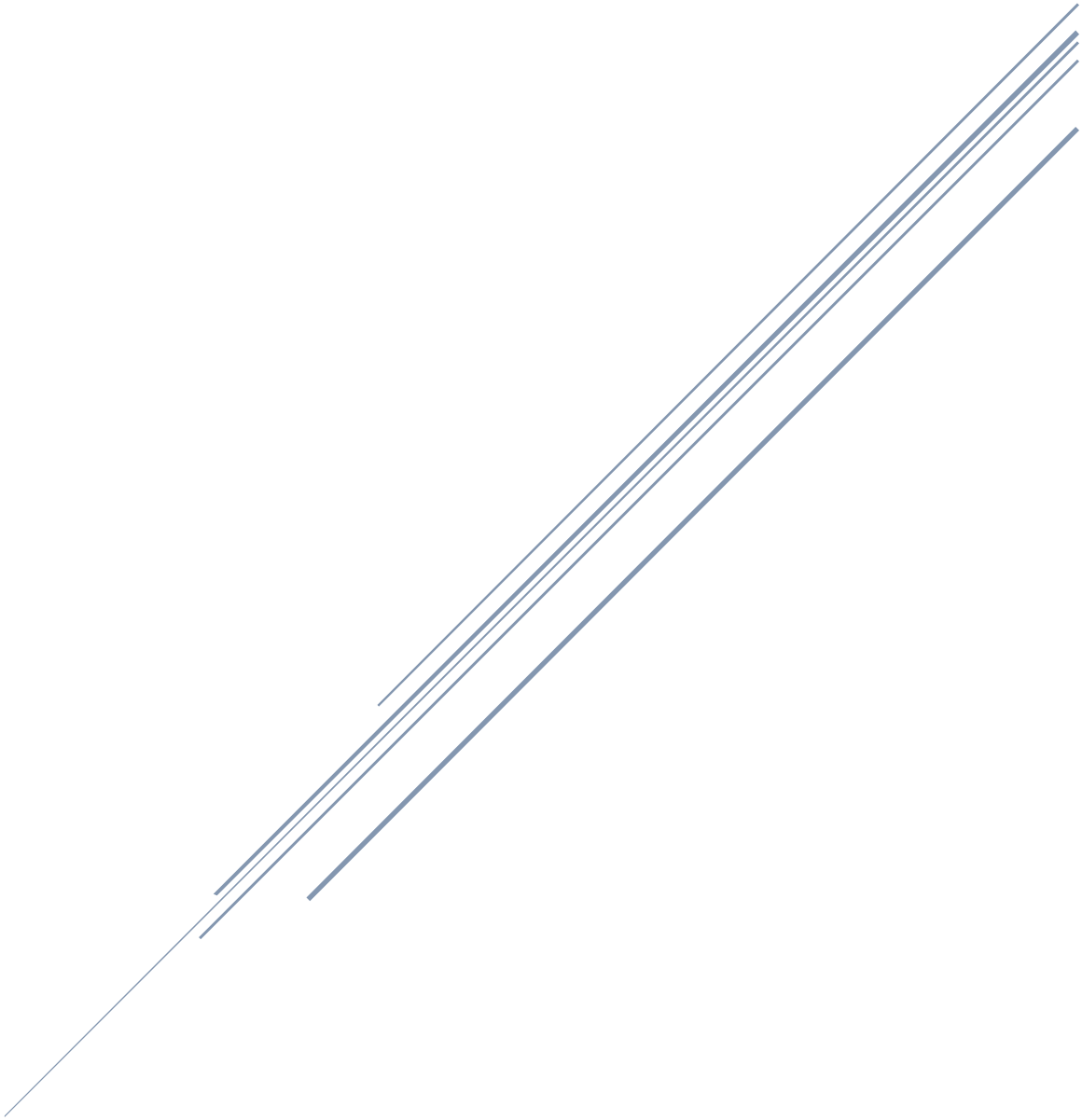
Assignment on Loop Statement

Course Title: Database Management System

Course Code: CSE-2203



Name: Md. Waliul Islam Rayhan
ID: B190305034

SQL Loop Statement

The LOOP statement allows to execute one or more statements repeatedly.

Here is the basic syntax of the LOOP statement:

```
[begin_label:] LOOP
statement_list
END LOOP [end_label]
```

The LOOP can have optional labels at the beginning and end of the block.

The LOOP executes the statement_list repeatedly. The statement_list may have one or more statements, each terminated by a semicolon (;) statement delimiter.

Typically, we terminate the loop when a condition is satisfied by using the LEAVE statement.

This is the typical syntax of the LOOP statement used with LEAVE statement:

```
[label]: LOOP
    ...
    -- terminate the loop
    IF condition THEN
        LEAVE [label];
    END IF;
    ...
END LOOP;
```

The LEAVE statement immediately exits the loop. It works like the break statement in other programming languages like C/C++, and Java.

In addition to the LEAVE statement, we can use the ITERATE statement to skip the current loop iteration and start a new iteration. The ITERATE is like the continue statement in PHP, C/C++, and Java.

Example

The following statement creates a stored procedure that uses a LOOP statement:

```
DROP PROCEDURE LoopDemo;

DELIMITER $$
CREATE PROCEDURE LoopDemo()
BEGIN
        DECLARE x  INT;
        DECLARE str  VARCHAR(255);

        SET x = 1;
        SET str =  '';

        loop_label:  LOOP
                IF  x > 10 THEN
                        LEAVE  loop_label;
                END  IF;

                SET  x = x + 1;
                IF  (x mod 2) THEN
                        ITERATE  loop_label;
                ELSE
                        SET  str = CONCAT(str,x,',');
                END  IF;
        END LOOP;
        SELECT str;
END$$
DELIMITER ;
```

In this example:

1. The stored procedure constructs a string from the even numbers e.g., 2, 4, and 6.
2. The loop_label before the LOOP statement for using with the ITERATE and LEAVE statements.
3. If the value of x is greater than 10, the loop is terminated because of the LEAVE statement.
4. If the value of the x is an odd number, the ITERATE ignores everything below it and starts a new loop iteration.
5. If the value of the x is an even number, the block in the ELSE statement will build the result string from even numbers.

The following statement calls the stored procedure:

```
CALL LoopDemo();
```

Here is the output:

```
+-------------+
| str         |
+-------------+
| 2,4,6,8,10, |
+-------------+
1 row in set (0.01 sec)


Query OK, 0 rows affected (0.02 sec)
```

So, we have learned how to use the MySQL LOOP statement to execute a block of code repeatedly based on a condition.

Thank you