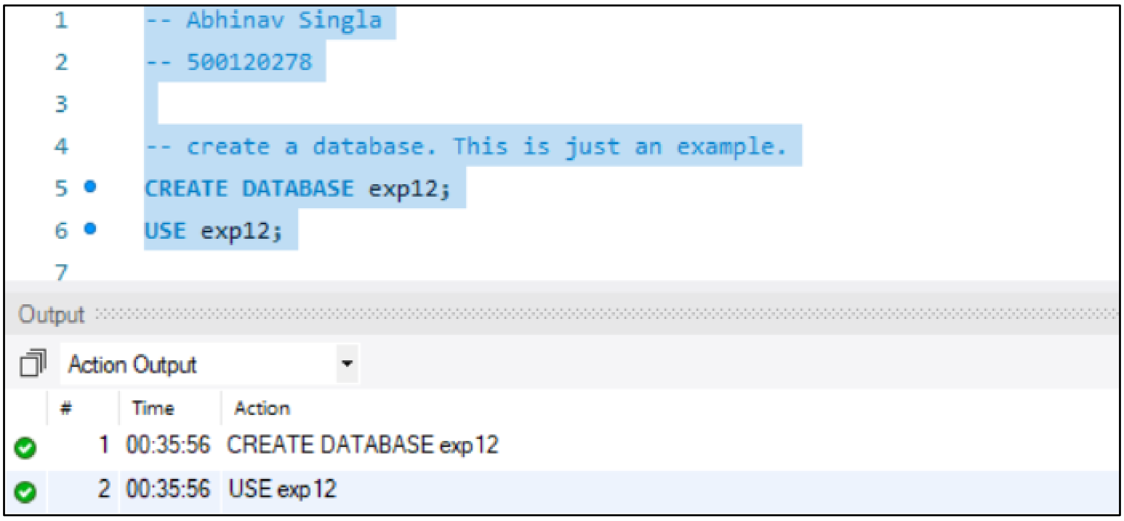
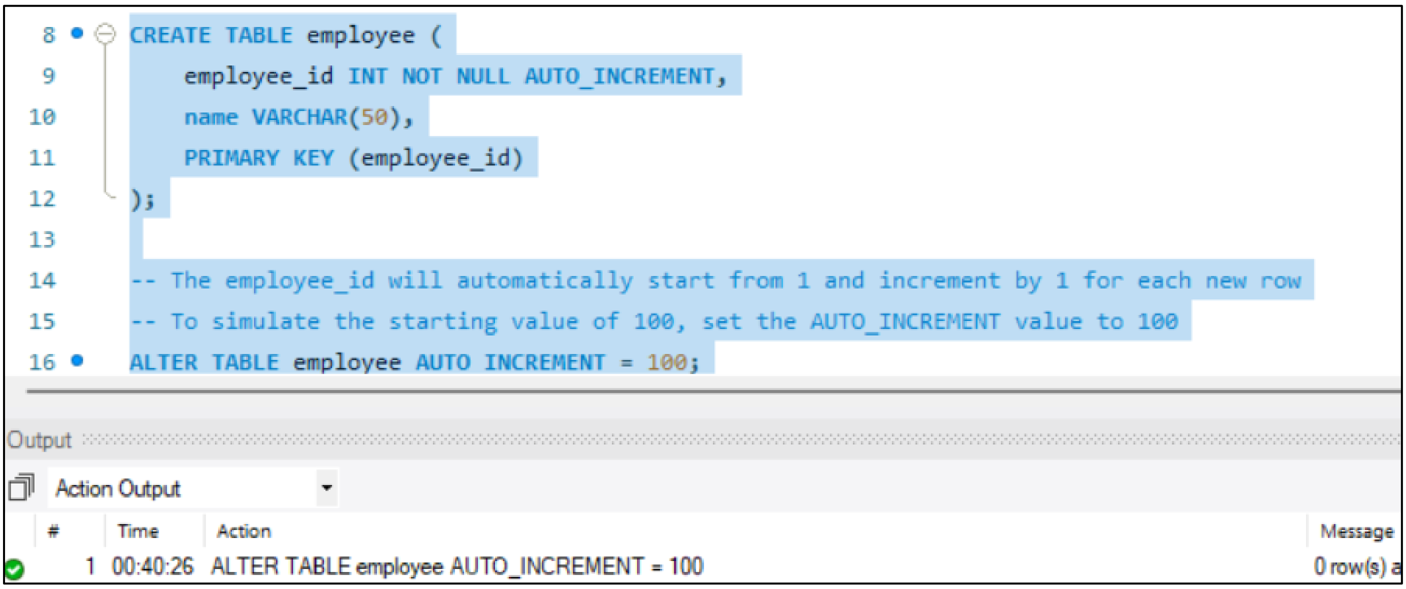
Title: To understand the concepts of Sequence. Objective: Students will be able to implement the concept of sequence.



```
-- Abhinav Singla
-- 500120278

-- create a database. This is just an example.
CREATE DATABASE exp12;
USE exp12;

```

```
Output
Action Output
  #   Time      Action
  1   00:35:56  CREATE DATABASE exp12
  2   00:35:56  USE exp12
```

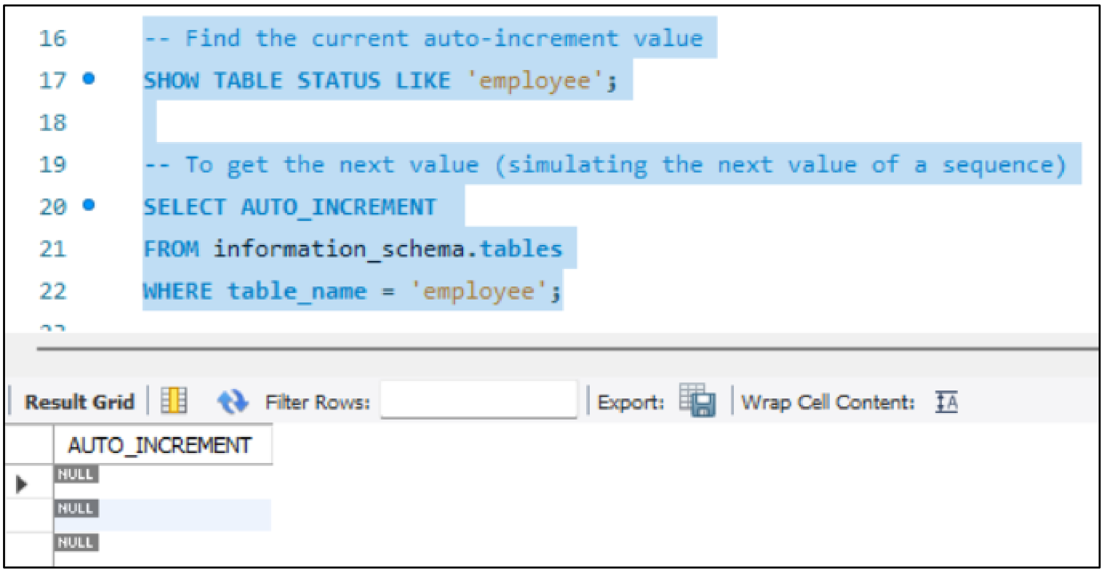
1) Create a sequence by name EMPID_SEQ starting with value 100 with an interval of 1.



```
CREATE TABLE employee (
    employee_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50),
    PRIMARY KEY (employee_id)
);

-- The employee_id will automatically start from 1 and increment by 1 for each new row
-- To simulate the starting value of 100, set the AUTO_INCREMENT value to 100
ALTER TABLE employee AUTO INCREMENT = 100;
```

```
Output
Action Output
  #   Time      Action                                          Message
  1   00:40:26  ALTER TABLE employee AUTO_INCREMENT = 100       0 row(s) a
```

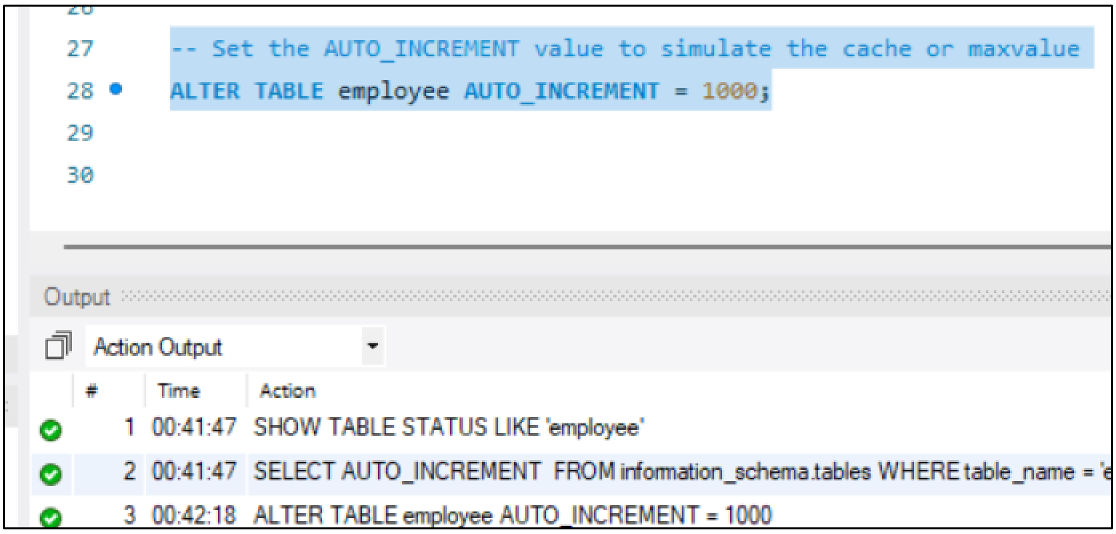
2) Write a SQL command for finding the current and the next status of EMPID_SEQ.



```sql
-- Find the current auto-increment value
SHOW TABLE STATUS LIKE 'employee';

-- To get the next value (simulating the next value of a sequence)
SELECT AUTO_INCREMENT
FROM information_schema.tables
WHERE table_name = 'employee';
```

```
AUTO_INCREMENT
NULL
NULL
NULL
```

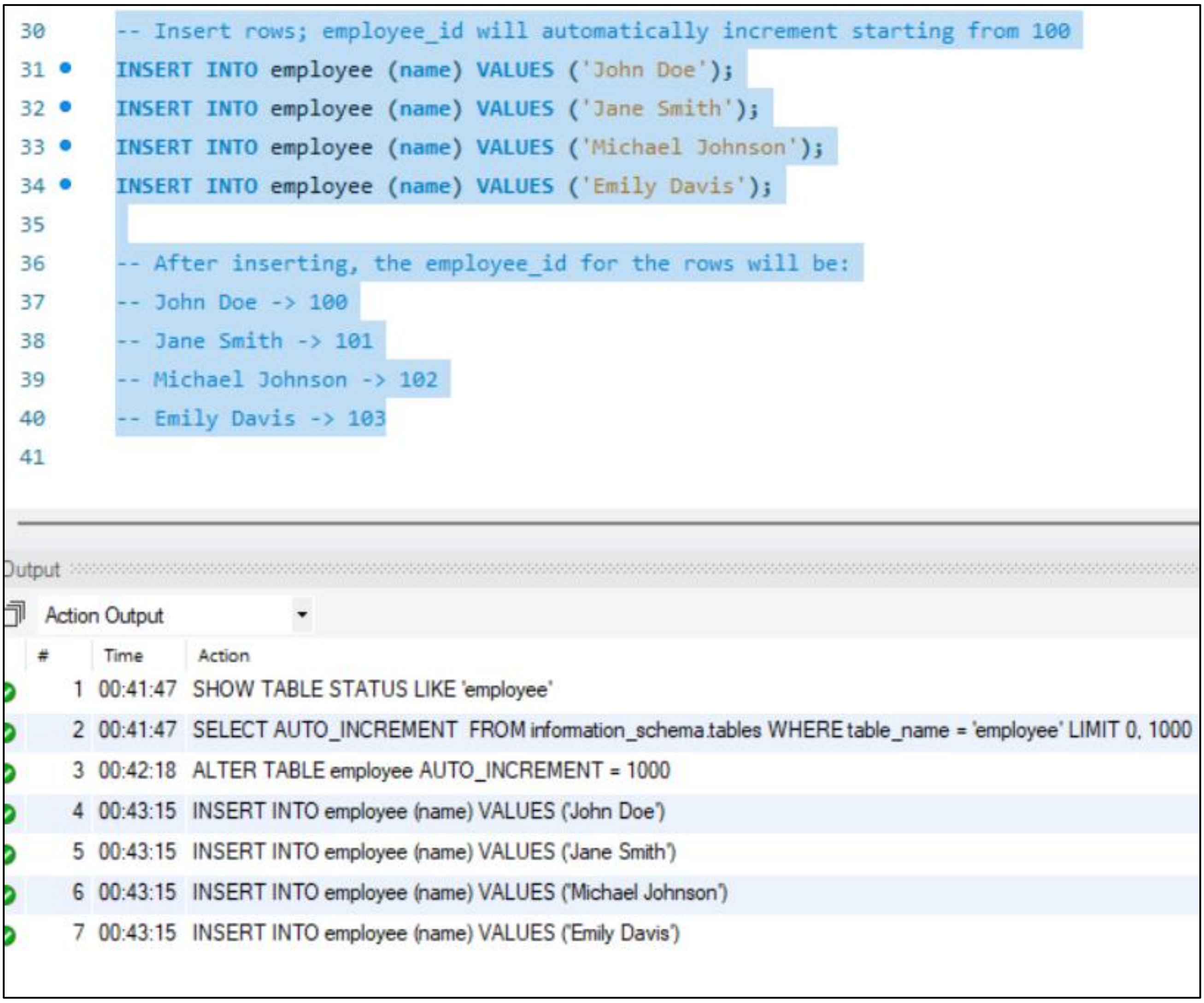
3) Change the Cache value of the sequence EMPID_SEQ to 20 and maxvalue to 1000.



```sql
-- Set the AUTO_INCREMENT value to simulate the cache or maxvalue
ALTER TABLE employee AUTO_INCREMENT = 1000;
```

```
#  Time      Action
1  00:41:47  SHOW TABLE STATUS LIKE 'employee'
2  00:41:47  SELECT AUTO_INCREMENT  FROM information_schema.tables WHERE table_name = 'e
3  00:42:18  ALTER TABLE employee AUTO_INCREMENT = 1000
```

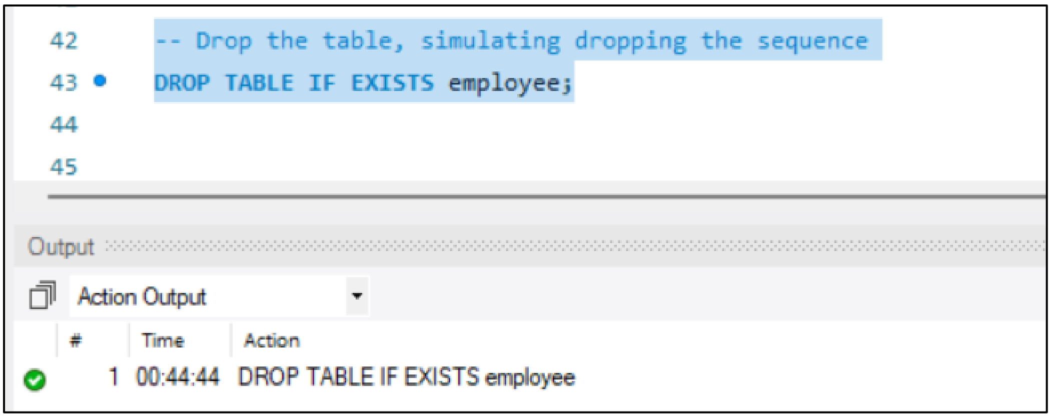
4) Insert values in employees table using sequences for employee_id column.



```
-- Insert rows; employee_id will automatically increment starting from 100
INSERT INTO employee (name) VALUES ('John Doe');
INSERT INTO employee (name) VALUES ('Jane Smith');
INSERT INTO employee (name) VALUES ('Michael Johnson');
INSERT INTO employee (name) VALUES ('Emily Davis');

-- After inserting, the employee_id for the rows will be:
-- John Doe -> 100
-- Jane Smith -> 101
-- Michael Johnson -> 102
-- Emily Davis -> 103
```

```
#  Time      Action
1  00:41:47  SHOW TABLE STATUS LIKE 'employee'
2  00:41:47  SELECT AUTO_INCREMENT  FROM information_schema.tables WHERE table_name = 'employee' LIMIT 0, 1000
3  00:42:18  ALTER TABLE employee AUTO_INCREMENT = 1000
4  00:43:15  INSERT INTO employee (name) VALUES ('John Doe')
5  00:43:15  INSERT INTO employee (name) VALUES ('Jane Smith')
6  00:43:15  INSERT INTO employee (name) VALUES ('Michael Johnson')
7  00:43:15  INSERT INTO employee (name) VALUES ('Emily Davis')
```

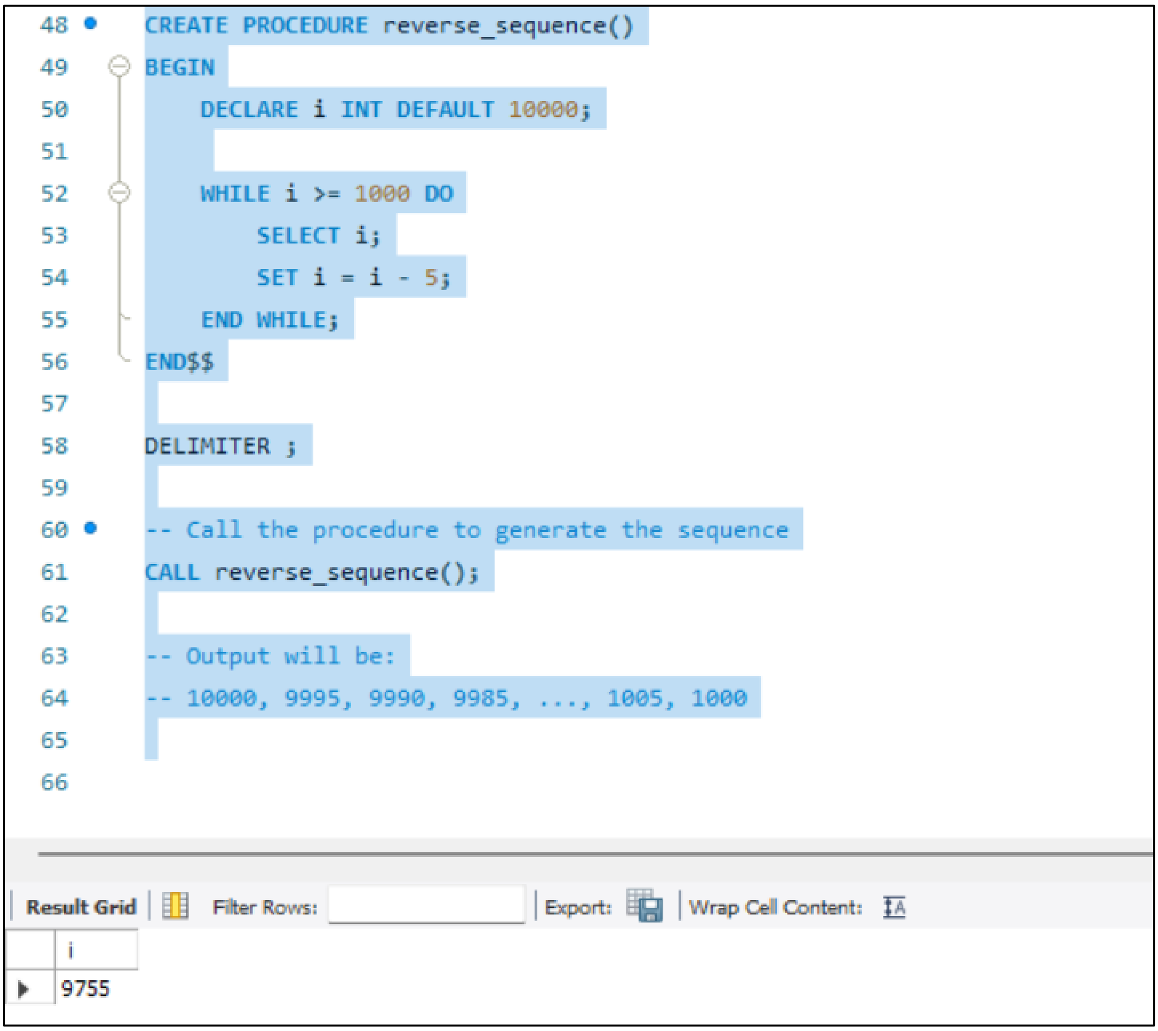
5) Drop sequence EMPID_SEQ.



```
-- Drop the table, simulating dropping the sequence
DROP TABLE IF EXISTS employee;
```

```
#  Time      Action
1  00:44:44  DROP TABLE IF EXISTS employee
```

6) Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

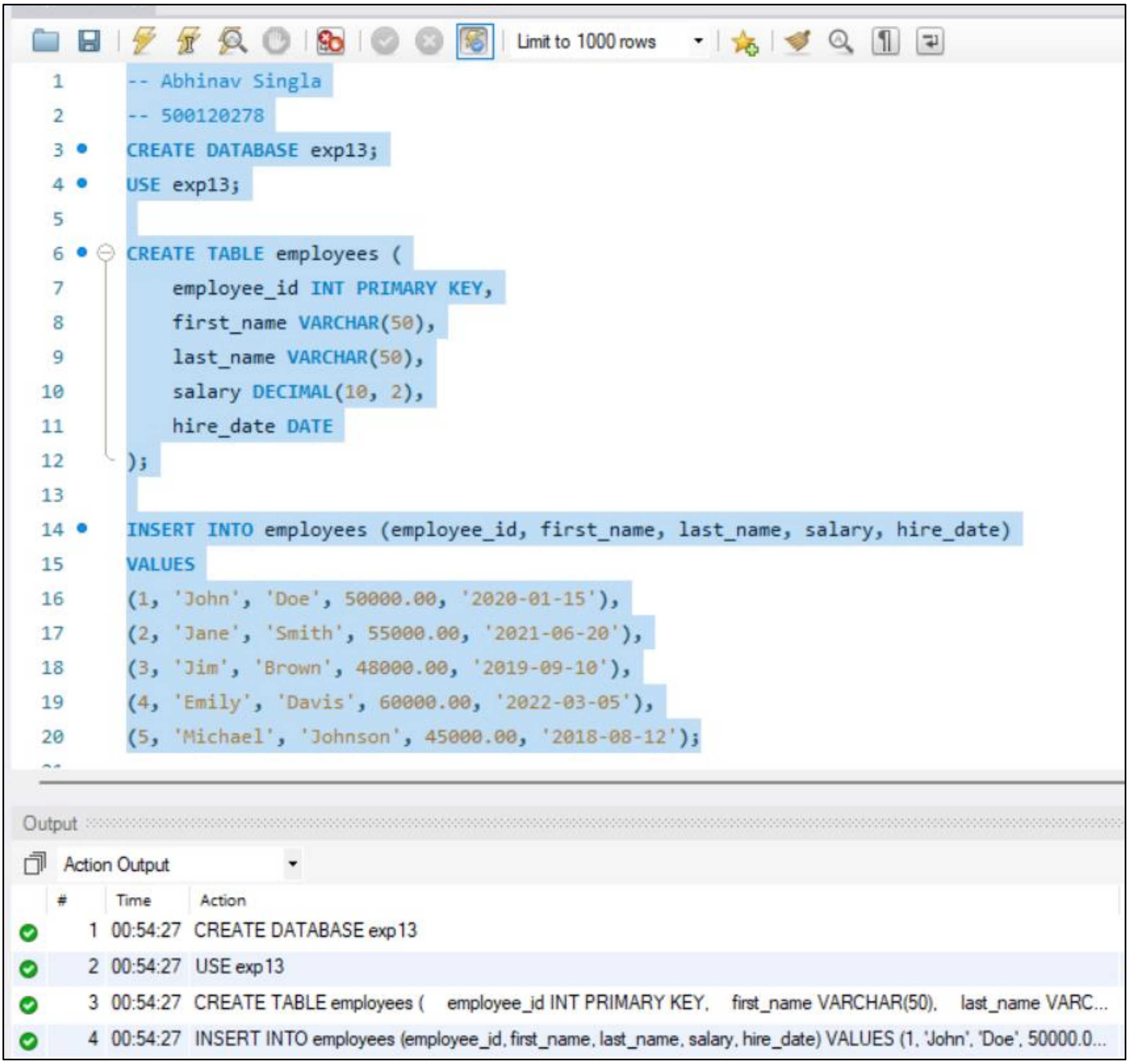


```
CREATE PROCEDURE reverse_sequence()
BEGIN
    DECLARE i INT DEFAULT 10000;

    WHILE i >= 1000 DO
        SELECT i;
        SET i = i - 5;
    END WHILE;
END$$

DELIMITER ;

-- Call the procedure to generate the sequence
CALL reverse_sequence();

-- Output will be:
-- 10000, 9995, 9990, 9985, ..., 1005, 1000
```

	i
▶	9755

Title: To understand the concepts of PL/SQL programming.
Objective: Students will be able to implement the basic concepts of Pl/SQL.

Creating the database



```sql
-- Abhinav Singla
-- 500120278
CREATE DATABASE exp13;
USE exp13;

CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    salary DECIMAL(10, 2),
    hire_date DATE
);

INSERT INTO employees (employee_id, first_name, last_name, salary, hire_date)
VALUES
(1, 'John', 'Doe', 50000.00, '2020-01-15'),
(2, 'Jane', 'Smith', 55000.00, '2021-06-20'),
(3, 'Jim', 'Brown', 48000.00, '2019-09-10'),
(4, 'Emily', 'Davis', 60000.00, '2022-03-05'),
(5, 'Michael', 'Johnson', 45000.00, '2018-08-12');
```

	#	Time	Action
✓	1	00:54:27	CREATE DATABASE exp13
✓	2	00:54:27	USE exp13
✓	3	00:54:27	CREATE TABLE employees (employee_id INT PRIMARY KEY, first_name VARCHAR(50), last_name VARC...
✓	4	00:54:27	INSERT INTO employees (employee_id, first_name, last_name, salary, hire_date) VALUES (1, 'John', 'Doe', 50000.0...

1) Write a PL/SQL code to accept the value of A, B & C display which is greater.
2) Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.
3) Write a PL/SQL code block to find the factorial of a number.
4) Write a PL/SQL program to generate Fibonacci series.
5) Write a PL/SQL code to fund the sum of first N numbers