3. Stored Procedure Using a Sequence

This example demonstrates using a stored procedure to generate a candidate ID and insert data into the table.

Procedure: `ADDCANDIDATE`

```sql
CREATE PROCEDURE ADDCANDIDATE (@N AS VARCHAR(50)) AS BEGIN DECLARE @A AS INT; -- Declare an integer to store the sequence value
DECLARE @C AS CHAR(5); -- Declare a character variable to store the ID SET @A = (NEXT VALUE FOR MYSEQ); -- Fetch the next value
from the sequence -- Conditional logic to format the ID based on the sequence value IF @A < 10 SET @C = CONCAT('C00', @A); --
Prefix with 'C00' for single-digit IDs ELSE IF @A < 100 SET @C = CONCAT('C0', @A); -- Prefix with 'C0' for two-digit IDs ELSE IF
@A < 1000 SET @C = CONCAT('C', @A); -- Prefix with 'C' for three-digit IDs -- Insert the formatted ID and name into the table
INSERT INTO CANDIDATE VALUES (@C, @N); END;
```

Key Points in the Procedure:

1. **Sequence Integration**: Uses `NEXT VALUE FOR MYSEQ` to fetch the next unique value.
2. **ID Formatting**: The ID is formatted with a prefix (`C00`, `C0`, `C`) to maintain a standard structure.
3. **Insert Operation**: The procedure inserts the formatted ID and the candidate's name into the `CANDIDATE` table.

Example Usage of the Procedure

```sql
EXEC ADDCANDIDATE 'John Doe'; EXEC ADDCANDIDATE 'Jane Smith';
```

Result in the `CANDIDATE` Table:

Candidate_ID	Name
C001	John Doe
C002	Jane Smith

Real-Life Scenario:

This type of sequence and procedure can be used in:

- **Employee Management Systems**: To generate unique employee IDs.
- **Student Enrollment**: To generate roll numbers for students.
- **Customer Management**: To generate customer account numbers.