

Full Code for the Function

sql

Copy code

```
CREATE FUNCTION GENID(@C CHAR(1), @I INT) RETURNS CHAR(5) AS BEGIN -- Declare variables DECLARE @R CHAR(5); DECLARE @ID CHAR(5); -- Use CASE to create the formatted ID SELECT @R = CASE WHEN @I < 10 THEN CONCAT(@C, '000', @I) WHEN @I < 100 THEN CONCAT(@C, '00', @I) WHEN @I < 1000 THEN CONCAT(@C, '0', @I) WHEN @I < 10000 THEN @C + CAST(@I AS CHAR(4)) ELSE 'NULL' END; -- Final formatted ID SET @ID = @R; -- Return the ID RETURN @ID; END;
```

Step-by-Step Explanation with Table

The function uses two inputs:

- 1. @C : A single character prefix (e.g., 'C').
- 2. @I : A number that is part of the ID.

Based on the value of @I , the function creates a 5-character string by padding zeros.

Input - @C	Input - @I	Condition	Operation	Result
'C'	5	@I < 10	Add '000' before the number	'C0005'
'C'	45	@I >= 10 AND @I < 100	Add '00' before the number	'C0045'
'C'	456	@I >= 100 AND @I < 1000	Add '0' before the number	'C0456'
'C'	1234	@I >= 1000 AND @I < 10000	Combine prefix with the number (no padding)	'C1234'
'C'	10000	@I >= 10000	Return 'NULL'	'NULL'

Breakdown of Each Step

1. Declaring Variables

sql

Copy code

```
DECLARE @R CHAR(5); DECLARE @ID CHAR(5);
```

- @R : Temporarily holds the formatted ID.
- @ID : Holds the final ID to return.

2. Using CASE to Pad Zeros

The logic for padding zeros is handled by this CASE block:

sql

Copy code

```
SELECT @R = CASE WHEN @I < 10 THEN CONCAT(@C, '000', @I) WHEN @I < 100 THEN CONCAT(@C, '00', @I)
WHEN @I < 1000 THEN CONCAT(@C, '0', @I) WHEN @I < 10000 THEN @C + CAST(@I AS CHAR(4)) ELSE 'NULL'
END;
```

- For @I < 10 : Adds '000' before the number.
- For @I < 100 : Adds '00' before the number.
- For @I < 1000 : Adds '0' before the number.
- For @I < 10000 : No padding, just combines the prefix with the number.
- For @I >= 10000 : Returns 'NULL' .

3. Assigning the Final ID

sql

Copy code

```
SET @ID = @R;
```

This step assigns the result of @R to the @ID variable.

4. Returning the Result

sql

Copy code

```
RETURN @ID;
```

The function returns the formatted ID.

Example Outputs

Input: @C	Input: @I	Formatted ID
'A'	1	'A0001'
'B'	12	'B0012'
'C'	123	'C0123'
'D'	4567	'D4567'

Input: @C	Input: @I	Formatted ID
'E'	10000	'NULL'

Explanation with Table

Let's break down an example where @C = 'C' and different values of @I are used.

Step	@C	@I	Condition	Operation	@R (Intermediate)	@ID (Final)
Input Values	'C'	5	@I < 10	Add '000' before the number	'C0005'	'C0005'
Input Values	'C'	45	@I >= 10 AND @I < 100	Add '00' before the number	'C0045'	'C0045'
Input Values	'C'	123	@I >= 100 AND @I < 1000	Add '0' before the number	'C0123'	'C0123'
Input Values	'C'	4567	@I >= 1000 AND @I < 10000	Combine prefix with the number (no padding)	'C4567'	'C4567'
Input Values	'C'	10000	@I >= 10000	Return 'NULL'	'NULL'	'NULL'

Key Points

- 1. **Handles Small Numbers:** Automatically pads smaller numbers with zeros.
- 2. **Ensures Fixed Length:** Output always has 5 characters unless the input exceeds 9999.
- 3. **Returns NULL for Large Numbers:** Prevents IDs from exceeding the character limit.

Let me know if you need further clarification! 😊