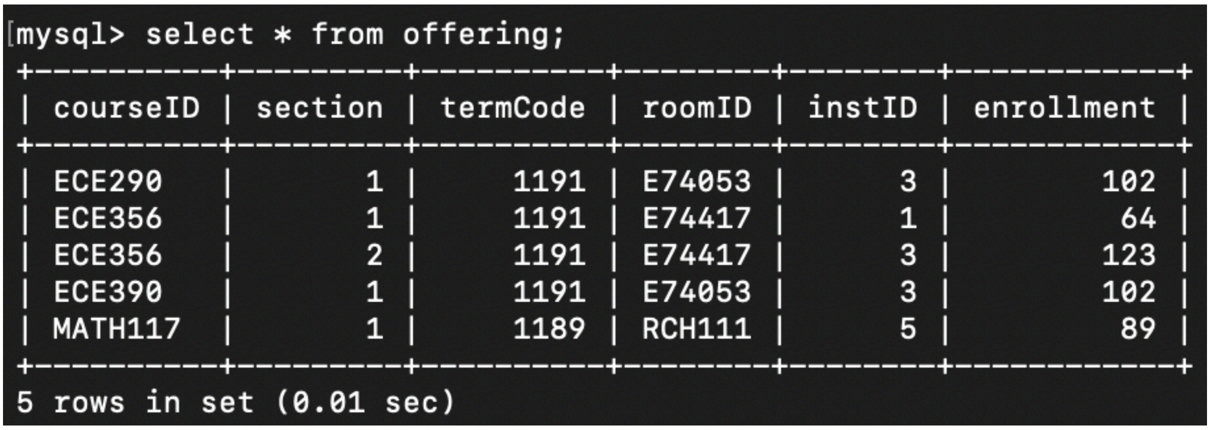
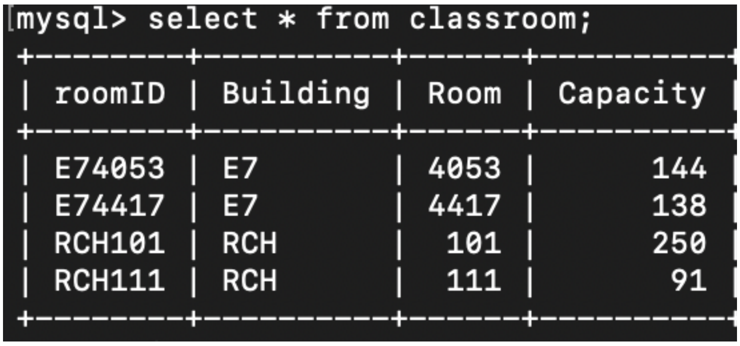
2.



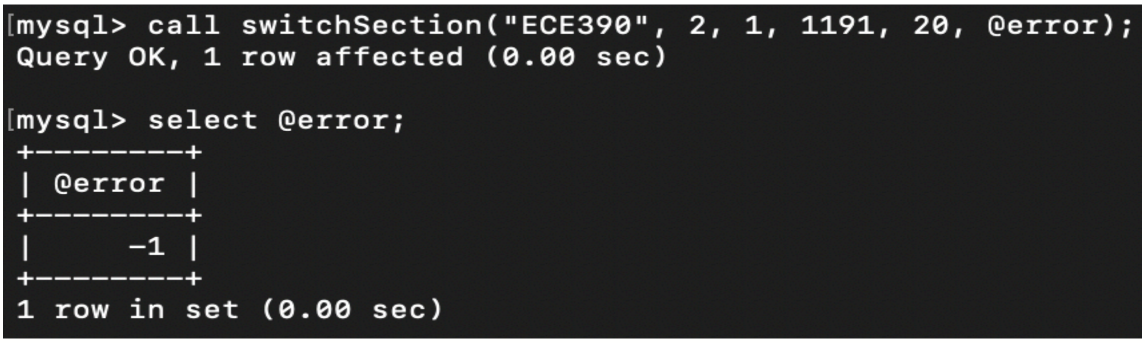
```
[mysql> select * from offering;
+----------+---------+----------+--------+--------+------------+
| courseID | section | termCode | roomID | instID | enrollment |
+----------+---------+----------+--------+--------+------------+
| ECE290   |       1 |     1191 | E74053 |      3 |        102 |
| ECE356   |       1 |     1191 | E74417 |      1 |         64 |
| ECE356   |       2 |     1191 | E74417 |      3 |        123 |
| ECE390   |       1 |     1191 | E74053 |      3 |        102 |
| MATH117  |       1 |     1189 | RCH111 |      5 |         89 |
+----------+---------+----------+--------+--------+------------+
5 rows in set (0.01 sec)
```



```
[mysql> select * from classroom;
+--------+----------+------+----------
| roomID | Building | Room | Capacity 
+--------+----------+------+----------
| E74053 | E7       | 4053 |      144 
| E74417 | E7       | 4417 |      138 
| RCH101 | RCH      |  101 |      250 
| RCH111 | RCH      |  111 |       91 
+--------+----------+------+----------
```

Test cases:

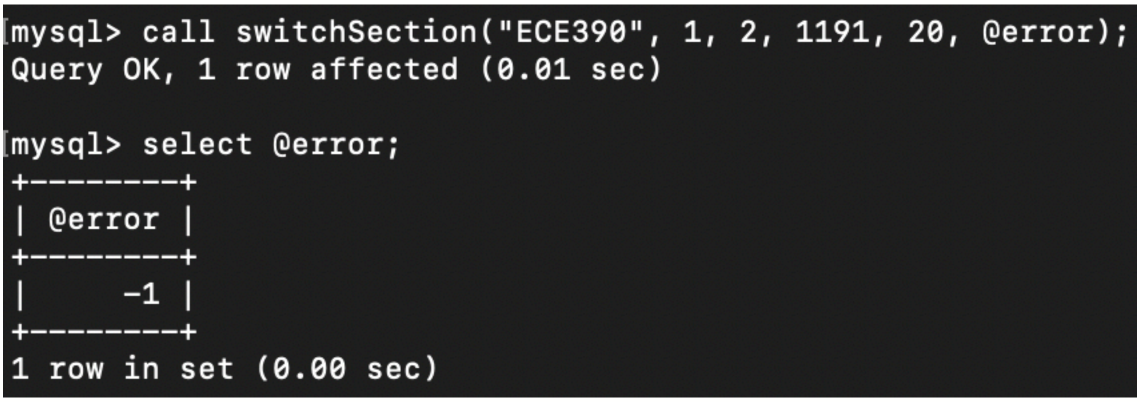
(courseID, section1, termCode) not in Offering



```
[mysql> call switchSection("ECE390", 2, 1, 1191, 20, @error);
Query OK, 1 row affected (0.00 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -1 |
+--------+
1 row in set (0.00 sec)
```

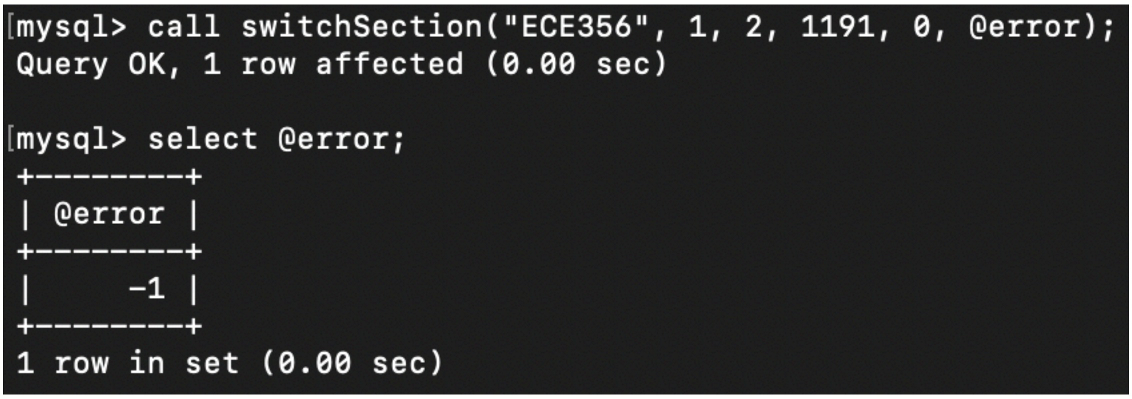
(courseID, section2, termCode) not in Offering



```
[mysql> call switchSection("ECE390", 1, 2, 1191, 20, @error);
Query OK, 1 row affected (0.01 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -1 |
+--------+
1 row in set (0.00 sec)
```

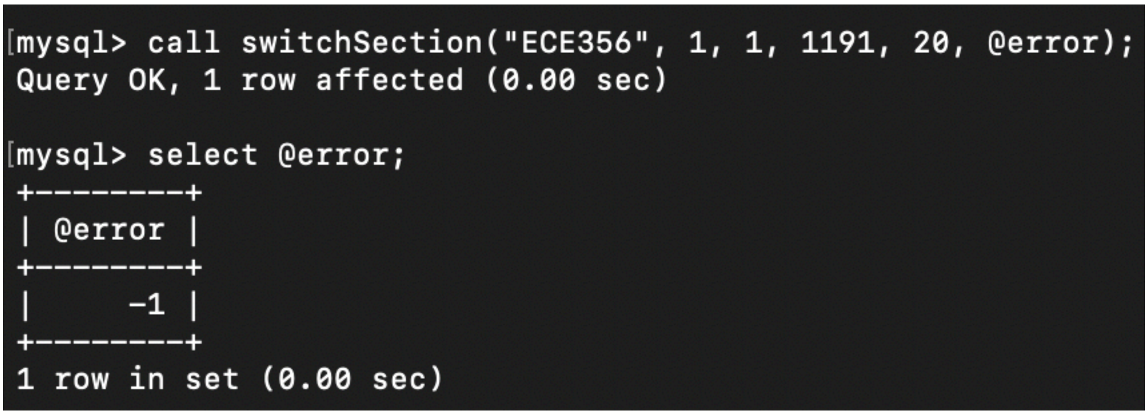
Quantity = 0



```
[mysql> call switchSection("ECE356", 1, 2, 1191, 0, @error);
Query OK, 1 row affected (0.00 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -1 |
+--------+
1 row in set (0.00 sec)
```

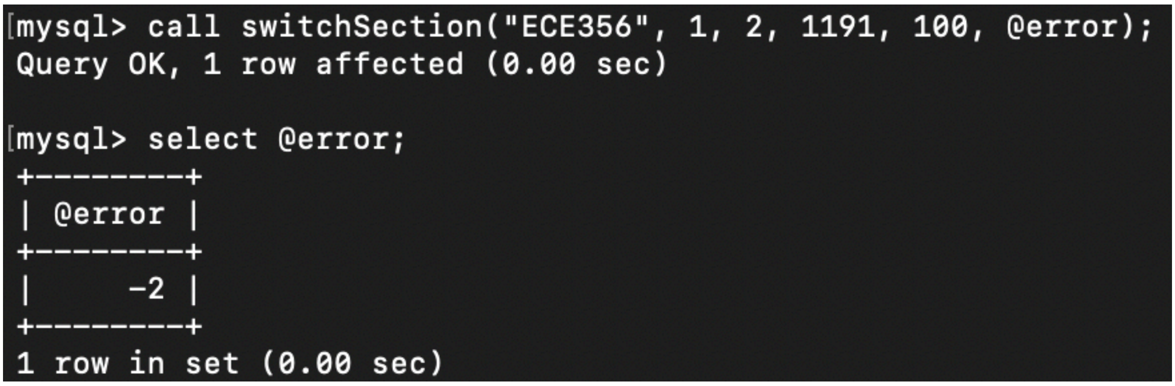
Section1 = Section2



```
[mysql> call switchSection("ECE356", 1, 1, 1191, 20, @error);
Query OK, 1 row affected (0.00 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -1 |
+--------+
1 row in set (0.00 sec)
```

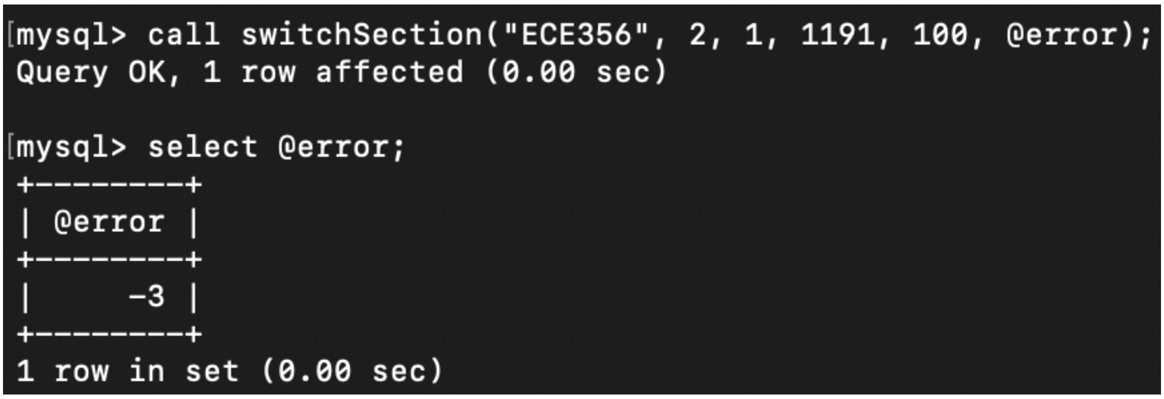
Decrease the enrollment in section1 by quantity, the enrollment is negative



```
[mysql> call switchSection("ECE356", 1, 2, 1191, 100, @error);
Query OK, 1 row affected (0.00 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -2 |
+--------+
1 row in set (0.00 sec)
```

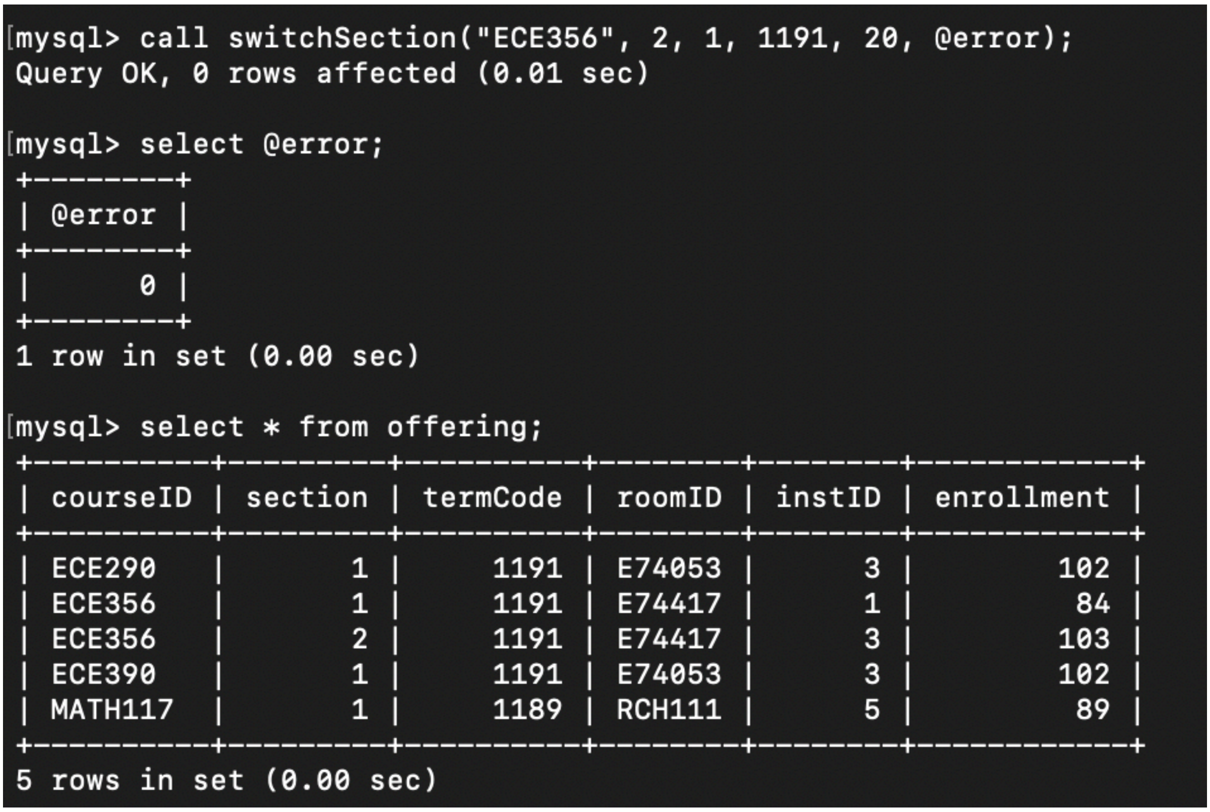
Increase the enrollment in section2 by quantity, the enrollment exceeds room capacity



```
[mysql> call switchSection("ECE356", 2, 1, 1191, 100, @error);
Query OK, 1 row affected (0.00 sec)

[mysql> select @error;
+--------+
| @error |
+--------+
|     -3 |
+--------+
1 row in set (0.00 sec)
```

No errors



```
[mysql> call switchSection("ECE356", 2, 1, 1191, 20, @error);
 Query OK, 0 rows affected (0.01 sec)

[mysql> select @error;
 +--------+
 | @error |
 +--------+
 |      0 |
 +--------+
 1 row in set (0.00 sec)

[mysql> select * from offering;
 +----------+---------+----------+--------+--------+------------+
 | courseID | section | termCode | roomID | instID | enrollment |
 +----------+---------+----------+--------+--------+------------+
 | ECE290   |       1 |     1191 | E74053 |      3 |        102 |
 | ECE356   |       1 |     1191 | E74417 |      1 |         84 |
 | ECE356   |       2 |     1191 | E74417 |      3 |        103 |
 | ECE390   |       1 |     1191 | E74053 |      3 |        102 |
 | MATH117  |       1 |     1189 | RCH111 |      5 |         89 |
 +----------+---------+----------+--------+--------+------------+
 5 rows in set (0.00 sec)
```

Can you achieve the same effect as the stored procedure by using check and/or trigger?

Within a stored function or trigger, it is not permitted to modify a table that is already being used (for reading or writing) by the statement that invoked the function or trigger.

For example, we set trigger that before/after we change the enrollment of ECE356 section2 by 20 in offering table, then we change the enrollment of ECE356 section1 by 20; however, the two changes are in the same column of the same table and trigger will occur on each row occurring updates, when the trigger leads to change the section1, it goes into an loop state, continuing update ECE356 section1 by 20 until not meet the condition and exit; therefore, we cannot achieve the same effect as the stored procedure.