

Explanation A

Explanation B

Explanation C

The method *parseSQL(query)* returns an SQL expression parsed for a given *query*.

In line 4, the code obtains the SQL *keyword* of the expression returned by *parseSQL(query)*. The assertion in line 6 checks if *keyword* has the same value as the variable *expected*.

Internally, in line 11, the method computes the value of *keyword* as a function of the input *query*. However,

- In V1, if the value of *keyword* is equal to “select” (line 12), the variable is reassigned to its uppercase version, i.e., “*SELECT*” (line 13).
- In V2, lines 12 and 13 are deleted and, thus, *keyword* is not reassigned to its uppercase version.

Then, in both versions, the variable *keyword* is appended to the variable *expr* (line 14). Next, *expr* is further appended with *function*, *table*, and *predicate* in lines 16, 18, and 20, respectively.

In summary, the deletion of lines 12-13 leads to the difference in the value of *keyword* in V1 and V2.

This difference causes the assertion to fail.

The method *parseSQL(query)* returns an SQL expression parsed for a given *query*. The input *query* is initialized in line 2 as “select sum(c1) from sales where c2>1”.

In line 4, the code obtains the SQL *keyword* of the expression returned by *parseSQL(query)*. The assertion in line 6 checks if *keyword* has the same value as the variable *expected*, which is set in line 5 to be “SELECT”.

Internally, in line 11, the method computes the value of *keyword* as a function of the input *query*. However,

- In V1, if the value of *keyword* is equal to “select” (line 12), the variable is reassigned to its uppercase version, i.e., “*SELECT*” (line 13).
- In V2, lines 12 and 13 are deleted and, thus, *keyword* is not reassigned to its uppercase version.

Then, in both versions, the variable *expr* is computed as a function of both *keyword* and *query* (line 20).

In summary, the deletion of lines 12-13 leads to the difference in the value of *keyword* in V1 and V2, for queries with *keyword* = “select”.

This difference causes the assertion to fail.

The method *parseSQL(query)* returns an SQL expression parsed for a given *query*.

In line 4, the code obtains the SQL *keyword* of the expression returned by *parseSQL(query)*. The assertion in line 6 checks if *keyword* has the same value as the variable *expected*.

Internally, in line 11, the method computes the value of *keyword* as a function of the input *query*. However,

- In V1, if the value of *keyword* is equal to “select” (line 12), the variable is reassigned to its uppercase version, i.e., “*SELECT*” (line 13).
- In V2, lines 12 and 13 are deleted and, thus, *keyword* is not reassigned to its uppercase version.

Then, in both versions, the variable *expr* is computed as a function of both *keyword* and *query* (line 20).

In summary, the deletion of lines 12-13 leads to the difference in the value of *keyword* in V1 and V2.

This difference causes the assertion to fail.

Notations: Colored backgrounds highlight the differences between the views.

FYI: Views are given below again, for your reference.

View A

View B

View C

V1		V2	
1	public static void main(String[] args){	1	public static void main(String[] args){
2		2	
3	SQLExpr expr = parseSQL(query)	3	SQLExpr expr = parseSQL(query)
4	String result = expr.getSQLKeyword();	4	String result = expr.getSQLKeyword();
5		5	
6	✅ assertEquals(result, expected);	6	❌ assertEquals(result, expected);
7	}	7	}
8	public SQLExpr parseSQL(String query){	8	public SQLExpr parseSQL(String query){
9		9	
10		10	
11		11	
12	if (keyword.equals("select")) [true]	12	
13	keyword = keyword.toUpper();	13	
14	expr = expr.append(keyword);	14	expr = expr.append(keyword);
15		15	
16	expr = expr.append(function);	16	expr = expr.append(function);
17		17	
18	expr = expr.append(table);	18	expr = expr.append(table);
19		19	
20	expr = expr.append(predicate);	20	expr = expr.append(predicate);
21	return expr;	21	return expr;
22	}	22	}

V1		V2	
1	public static void main(String[] args){	1	public static void main(String[] args){
2	String query = "select sum(c1) from sales where c2>1";	2	String query = "select sum(c1) from sales where c2>1";
3	SQLExpr expr = parseSQL(query);	3	SQLExpr expr = parseSQL(query)
4	String result = expr.getSQLKeyword();	4	String result = expr.getSQLKeyword();
5	String expected = "SELECT";	5	String expected = "SELECT";
6	✅ assertEquals(result, expected);	6	❌ assertEquals(result, expected);
7	}	7	}
8	public SQLExpr parseSQL(String query){	8	public SQLExpr parseSQL(String query){
9		9	
10		10	
11	String keyword = Func1(query);	11	String keyword = Func1(query);
12	if (keyword.equals("select")) [true]	12	
13	keyword = keyword.toUpper();	13	
14		14	
15		15	
16		16	
17		17	
18		18	
19		19	
20	expr = Func2(keyword, query);	20	expr = Func2(keyword, query);
21	return expr;	21	return expr;
22	}	22	}

V1		V2	
1	public static void main(String[] args){	1	public static void main(String[] args){
2		2	
3	SQLExpr expr = parseSQL(query)	3	SQLExpr expr = parseSQL(query)
4	String result = expr.getSQLKeyword();	4	String result = expr.getSQLKeyword();
5		5	
6	✅ assertEquals(result, expected);	6	❌ assertEquals(result, expected);
7	}	7	}
8	public SQLExpr parseSQL(String query){	8	public SQLExpr parseSQL(String query){
9		9	
10		10	
11		11	
12	if (keyword.equals("select")) [true]	12	
13	keyword = keyword.toUpper();	13	
14		14	
15		15	
16		16	
17		17	
18		18	
19		19	
20	expr = Func2(keyword, query);	20	expr = Func2(keyword, query);
21	return expr;	21	return expr;
22	}	22	}