## Content Corrections

<table>
<thead>
<tr>
<th>Page Number</th>
<th>Item</th>
<th>Comment</th>
</tr>
</thead>
</table>
| 40          | Chapter 1, Chapter Quiz, Question 3 | The FROM clause should not have a semicolon at the end. The code should read as follows:  
`proc sql;  
  from certadv.size left join certadv.price  
on size.address = price.address;  
quit;` |
| 91          | Chapter 3, Chapter Quiz, Question 3 | The references to table1 and table2 should include the librefs and should read as follows:  
`proc sql;  
  select * from certadv.table1 left join certadv.table2  
on table1.g3=table2.g3;  
quit;` |
| 92          | Chapter 3, Chapter Quiz, Question 5 | SQL does not require that input data sets be sorted to merge them. However, when merging data through a DATA step, the input data sets must be sorted by the BY variable. To create the MERGED data set that is shown in the PROC PRINT output, the input data, certadv.table1 and certadv.table2, must first be sorted by the g3 variable. The code to create the MERGED data set that is referenced by the PROC PRINT step is as follows:  
`proc sort data=certadv.table1 out=table1;  
  by g3;  
run;  
proc sort data=certadv.table2 out=table2;  
  by g3;  
run;  
data merged;  
merge table1 table2;  
  by g3;  
run;  
proc print data=merged noobs;  
  title 'Merged';  
run;` |

In addition, the correct output from the PROC PRINT step is as follows:

```
G3  Z  R
46  UI  BC
47  BA
85  FL
89  FL
99  BA
```
The correct answer is C, as indicated in the Chapter Quiz Answer Key. The code as shown on page 93 is correct. However, a more robust solution is to replace the G3 label with a column named G3, as follows:

```sas
proc sql;
  title 'Merged';
  select coalesce(a.g3, b.g3) as G3, z, r
  from certadv.table1 as a
  full join certadv.table2 as b
  on a.g3 = b.g3
  order by 1;
  title;
quit;
```

The correct output is as follows:

<table>
<thead>
<tr>
<th>EmpID</th>
<th>JobCode</th>
<th>DateOfBirth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1574</td>
<td>FA2</td>
<td>01MAY1988</td>
</tr>
<tr>
<td>1125</td>
<td>FA2</td>
<td>12NOV1976</td>
</tr>
<tr>
<td>1475</td>
<td>FA2</td>
<td>19DEC1969</td>
</tr>
<tr>
<td>1124</td>
<td>FA1</td>
<td>13JUL1966</td>
</tr>
<tr>
<td>1422</td>
<td>FA1</td>
<td>08JUN1972</td>
</tr>
<tr>
<td>1094</td>
<td>FA1</td>
<td>05APR1978</td>
</tr>
<tr>
<td>1368</td>
<td>FA2</td>
<td>15JUN1969</td>
</tr>
<tr>
<td>1411</td>
<td>FA2</td>
<td>31MAY1969</td>
</tr>
<tr>
<td>1113</td>
<td>FA1</td>
<td>18JAN1976</td>
</tr>
<tr>
<td>1441</td>
<td>FA2</td>
<td>23NOV1977</td>
</tr>
<tr>
<td>1103</td>
<td>FA1</td>
<td>19FEB1976</td>
</tr>
<tr>
<td>1477</td>
<td>FA2</td>
<td>25MAR1972</td>
</tr>
<tr>
<td>1970</td>
<td>FA1</td>
<td>29SEP1972</td>
</tr>
<tr>
<td>1424</td>
<td>FA2</td>
<td>08AUG1977</td>
</tr>
<tr>
<td>1132</td>
<td>FA1</td>
<td>03JUN1980</td>
</tr>
<tr>
<td>1413</td>
<td>FA2</td>
<td>20SEP1973</td>
</tr>
<tr>
<td>1116</td>
<td>FA1</td>
<td>02OCT1977</td>
</tr>
<tr>
<td>1555</td>
<td>FA2</td>
<td>20MAR1976</td>
</tr>
<tr>
<td>1434</td>
<td>FA2</td>
<td>14OCT1970</td>
</tr>
<tr>
<td>1414</td>
<td>FA1</td>
<td>28MAR1970</td>
</tr>
<tr>
<td>1390</td>
<td>FA2</td>
<td>23FEB1973</td>
</tr>
<tr>
<td>1425</td>
<td>FA1</td>
<td>31DEC1979</td>
</tr>
<tr>
<td>1135</td>
<td>FA2</td>
<td>24SEP1968</td>
</tr>
<tr>
<td>1415</td>
<td>FA2</td>
<td>12MAR1966</td>
</tr>
<tr>
<td>1221</td>
<td>FA2</td>
<td>25SEP1975</td>
</tr>
<tr>
<td>1130</td>
<td>FA1</td>
<td>19MAY1979</td>
</tr>
<tr>
<td>1122</td>
<td>FA2</td>
<td>04MAY1971</td>
</tr>
<tr>
<td>Page</td>
<td>Chapter, Example</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 253  | Chapter 9, Example: Using the %DO Statement | The PUTLOOP macro should read as follows: 
```sas
%macro putloop;
  %local i;
  %do i=1 %to &sqlobs;
    %put TEACH\i is &&teach\i;
  %end;
%mend;
```
| 254  | Chapter 9, Example: Generating Complete Steps | The WHERE statement in the ROSTERS macro should end in a semicolon as follows: 
`where Course_Number=&class;`
| 373  | Chapter 16, Chapter Quiz Answer Keys, Chapter 1, Question 2 | Correct Answer: B  
Updated description: 
There are three statements, the PROC SQL statement, the SELECT statement, and the QUIT statement. The SELECT statement contains three clauses: the SELECT clause, the FROM clause, and the ORDER BY clause.
| 374  | Chapter 16, Chapter Quiz Answer Keys, Chapter 1, Question 6 | Delete "(optional)" from the explanation.
| 374  | Chapter 16, Chapter Quiz Answer Keys, Chapter 1, Question 7 | Correct Answer: C
| 374  | Chapter 16, Chapter Quiz Answer Keys, Chapter 2, Question 5 | Correct Answer: C
Creating New Columns Using the CASE Expression

New columns can also be created using the CASE expression if certain conditions are met. A CASE expression returns a single value that is conditionally evaluated for each row of a table (or view). A series of WHEN-THEN clauses are evaluated in sequence to find a match. The first WHEN clause that is evaluated as True determines which value the CASE expression returns. An optional ELSE expression gives an alternate action when no THEN expression is executed.

```sql
proc sql;
   select empid, salary,
       case
           when salary < 38823 then "low"
           when salary < 56592 then "mid"
           when salary >= 56592 then "high"
           else "error"
       end as salrange
   from certadv.payrollmaster;
quit;
```

The CASE expression above creates a new column named Salrange. In a row of data when salary is equal to 45000, the first WHEN clause evaluates as False. The second WHEN clause evaluates as True, and the value of "mid" is assigned to the new Salrange column. Because the second WHEN clause evaluated as True, the third WHEN clause and the ELSE clause do not evaluate.

When testing for equality, an abbreviated form of the CASE expression can be used:

```sql
proc sql;
   select aname, country,
       case country
           when "New Zealand" then "APAC"
           when "Netherlands" then "Europe"
           when "USA" then "Americas"
           else "Other"
       end as region
   from certadv.airports;
quit;
```