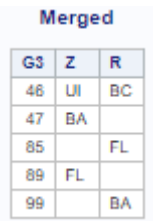*Font that is in italics is what was changed in the text, unless otherwise specified.

Content Corrections

Page Number	Item	Comment

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:
		
		Merged
		<table><tr><th>G3</th><th>Z</th><th>R</th></tr><tr><td>46</td><td>UI</td><td>BC</td></tr><tr><td>47</td><td>BA</td><td></td></tr><tr><td>85</td><td></td><td>FL</td></tr><tr><td>89</td><td>FL</td><td></td></tr><tr><td>99</td><td></td><td>BA</td></tr></table>

*Font that is in italics is what was changed in the text, unless otherwise specified.

| 93 | Chapter 3, Chapter Quiz, Question 5 | 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;
``` |
137	Chapter 5, Output 5.3	The correct output is as follows: | EmpID	JobCode	DateOfBirth
1574	FA2	01MAY1988
1125	FA2	12NOV1976
1475	FA2	19DEC1969
1124	FA1	13JUL1966
1422	FA1	08JUN1972
1094	FA1	05APR1978
1368	FA2	15JUN1969
1411	FA2	31MAY1969
1113	FA1	18JAN1976
1441	FA2	23NOV1977
1103	FA1	19FEB1976
1477	FA2	25MAR1972
1970	FA1	29SEP1972
1424	FA2	08AUG1977
1132	FA1	03JUN1980
1413	FA2	20SEP1973
1116	FA1	02OCT1977
1555	FA2	20MAR1976
1434	FA2	14JUL1970
1414	FA1	28MAR1970
1390	FA2	23FEB1973
1425	FA1	31DEC1979
1135	FA2	24SEP1968
1415	FA2	12MAR1966
1221	FA2	25SEP1975
1130	FA1	19MAY1979
1122	FA2	04MAY1971	

*Font that is in italics is what was changed in the text, unless otherwise specified.

149	Chapter 5, Chapter Quiz Question 1	The question should read: Which PROC SQL query *does not* remove duplicate values of MemberType from the query output, so that only the unique values are listed?

151	Chapter 5, Chapter Quiz Question 5	Answer d should read:
		
		``
		proc sql;
		    select name, checkedout
		        from certadv.circulation
		        where name in
		            (select name
		                from certadv.volunteers;);
		quit;
		``

151	Chapter 5, Chapter Quiz Question 7	The data sets, literacy and continents, that are referenced in the question are not included in the sample data. The code is provided for illustrative purposes only.

253	Chapter 9, Example: Using the %DO Statement	The PUTLOOP macro should read as follows:
		``
		%macro putloop;
		    %local i;
		    %do i=1 %to &sqlobs;
		        %put TEACH&i is &&teach&i;
		    %end;
		%mend;
		%putloop
		``

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

*Font that is in italics is what was changed in the text, unless otherwise specified.

374	Chapter 16, Chapter Quiz Answer Keys, Chapter 2, Question 5	Correct Answer: C

*Font that is in italics is what was changed in the text, unless otherwise specified.

Additional Content

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.

```
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:

```
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;
```