Using W3Schools.com and SQLZoo.Net as resources, complete the following lab.

Ray Henry is the owner of a bookstore chain named Henry Books. Ray has decided to store his data in a SQL 2005 database. He wants to ensure his data is safe, current, and accurate. He would also like to be able to ask questions concerning his data and have you obtain the answers for him quickly and concisely. A new database has been completed and all of his data has been imported into this new database. You; as the companies super database guru/king; are tasked with attaching this new database to your system and answering the questions Mr. Henry has asked.

Attach the HENRY database. Write the SQL query that answers the following scenarios.

1. List the book code and book title of each book. (33 Rows)

**select book\_code, book\_title**

**from books;**

**var query1 = from b in Books**

**select new { b.BOOK\_CODE, b.BOOK\_TOTAL };**

**query1.Dump();**

**var query1 = Books.Select(b => new {**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**});**

**query1.Dump();**

1. List the complete PUBLISHER table. (28 rows)

**select \* from publishers**

**var query2 = Publishers;**

**query2.Dump();**

**var query2 = publishers;**

**query2.Dump();**

1. List the name of each publisher located in Boston. (2 rows)

**select publisher\_name**

**from publishers**

**where publisher\_city = 'Boston';**

**select publisher\_name**

**from publishers**

**where publisher\_city in ('Boston');**

**var query3 = from p in Publishers**

**where p.PUBLISHER\_CITY = “Boston”**

**select new { p.PUBLISHER\_NAME, p.PUBLISHER\_CITY };**

**query3.Dump();**

**var query3 = publishers.Where(p => p.PUBLISHER\_CITY == “Boston”)**

**.Select(p => new**

**{**

**p.PUBLISHER\_NAME**

**p.PUBLISHER\_CITY**

**});**

**query3.Dump();**

1. List the name of each publisher not located in Boston. (26 rows)

**select publisher\_name**

**from publishers**

**where publisher\_city not in ('Boston');**

**select publisher\_name**

**from publishers**

**where publisher\_city <> ('Boston');**

**select publisher\_name**

**from publishers**

**where publisher\_city != ('Boston');**

**var query4 = from p in Publishers**

**where p.PUBLISHER\_CITY != “Boston”**

**orderby p.PUBLISHER\_CITY, p.PUBLISHER\_NAME**

**select new { p.PUBLISHER\_NAME, p.PUBLISHER\_CITY };**

**query4.Dump();**

**var query4 = publishers.Where(p => p.PUBLISHER\_CITY != “Boston”)**

**.Select(p => new**

**{**

**p.PUBLISHER\_NAME**

**p.PUBLISHER\_CITY**

**})**

**.OrderBy(p => p.PUBLISHER\_CITY);**

**query4.Dump();**

1. List the name of each branch that has at least nine employees. (3 rows)

**select branch\_name**

**from branches**

**where branch\_emps >= 9;**

**var query5 = from b in Branches**

**where b.BRANCH\_EMPS >= 9**

**orderby b.BRANCH\_EMPS**

**select new { b.BRANCH\_NAME, b.BRANCH\_EMPS };**

**query5.Dump();**

**var query5 = Branches.Where(b => b.BRANCH\_EMP >= 9)**

**.Select(b => b)**

**.OrderBy (b => b.BRANCH\_EMPS);**

**query5.Dump();**

1. List the book code and book title of each book that has the type SFI. (3 rows)

**select book\_code, book\_title**

**from books**

**where book\_type = 'sfi'**;

**var query6 = from b in Books**

**where b.BOOK\_TYPE == “SFI”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE };**

**query6.Dump();**

**var query6 = Books.Where(b => b.BOOK\_TYPE == “SFI”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE**

**});**

**query6.Dump();**

1. List the book code and book title of each book that has the type SFI and is in paperback. (1 row)

**select book\_code, book\_title**

**from books**

**where book\_type = 'sfi' and book\_paperback = 'Y';**

**var query7 = from b in Books**

**where b.BOOK\_TYPE == “SFI” &&**

**b.BOOK\_PAPERBACK == “Y”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE };**

**query7.Dump();**

**var query7 = Books.Where(b => b.BOOK\_TYPE == “SFI” &&**

**b.BOOK\_PAPERBACK == “Y”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE**

**});**

**query7.Dump();**

1. List the book code and book title of each book that has the type SFI or is published by the publisher with code SC. (5 rows)

publisher with code SC. (5 rows)

**select book\_code, book\_title**

**from books**

**where book\_type = 'sfi' or publisher\_code = 'sc';**

**var query8 = from b in Books**

**where b.BOOK\_TYPE == “SFI” ||**

**b.PUBLISHER\_CODE == “SC”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE };**

**query8.Dump();**

**var query8 = Books.Where(b => b.BOOK\_TYPE == “SFI” ||**

**b.PUBLISHER\_CODE == “SC”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE,**

**b.PUBLISHER\_CODE**

**});**

**query8.Dump();**

1. List the book code, book title, and price of each book with a price between $20 and $30. (4 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price >= 20 and book\_price <= 30;**

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price between 20 and 30;**

**var query9 = from b in Books**

**where b.BOOK\_PRICE >= 20m &&**

**b.BOOK\_PRICE <= 30m**

**orderby b.BOOK\_PRICE**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_PRICE };**

**query9.Dump();**

**var query9 = Books.Where(b => b.BOOK\_PRICE >= 20m &&**

**b.BOOK\_PRICE <= 30m)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE,**

**b.BOOK\_PRICE**

**});**

**query9.Dump();**

1. List the book code and book title of each book that has the type MYS and a price of less than $20. (2 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price < 20 and book\_type = 'mys';**

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price between 0.01 and 19.99 and book\_type in ('mys');**

**var query10 = from b in Books**

**where b.BOOK\_TYPE == “MYS” &&**

**b.BOOK\_PRICE < 20m**

**orderby b.BOOK\_PRICE**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_PRICE };**

**query10.Dump();**

**var query10 = Books.Where(b => b.BOOK\_TYPE == “MYS” &&**

**b.BOOK\_PRICE < 20m)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE,**

**b.BOOK\_PRICE**

**});**

**query10.Dump();**

1. Customers who are part of a special program get a 10% discount off regular book prices. List the book code, book title, and discounted price for each book. Use discounted\_price as the name for the computed column, which should calculate 90% of the current price; that is, 100% less a 10% discount. (33 rows)

**select book\_code, book\_title, book\_price, round(book\_price \* 0.9, 2) as discounted\_price**

**from books;**

**var query11 = from b in Books**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, Discounted\_Price = (b.BOOK\_PRICE \* 0.9m), b.BOOK\_PRICE };**

**query11.Dump();**

**var query11 = Books.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**Discounted\_Price = b.BOOK\_PRICE \* 0.9m**

**});**

**query11.Dump();**

1. Find the name of each publisher containing the word “and”. (4 rows)

**select publisher\_name**

**from publishers**

**where publisher\_name like '%and%';**

**var query12 = from p in Publishers**

**where p.PUBLISHER\_NAME.Contains(“and”)**

**select new { p.PUBLISHER\_NAME };**

**query12.Dump();**

**var query12 = Publishers.Where(p => p.PUBLISHER\_NAME.Contains(“and”))**

**.Select(p => p.PUBLISHER\_NAME);**

**query12.Dump();**

1. List the book code and book title of each book that has the type SFI, MYS, or ART. (9 rows)

**select book\_code, book\_title**

**from books**

**where book\_type in ('sfi', 'mys', 'art');**

**select book\_code, book\_title**

**from books**

**where book\_type = 'sfi' or book\_type = 'mys' or book\_type = 'art';**

**var query13 = from b in Books**

**orderby b.BOOK\_TYPE**

**where b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “ART”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE };**

**query13.Dump();**

**var query13 = Books.Where(b => b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “ART”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE**

**})**

**.OrderBy(b => b.BOOK\_TYPE);**

**query13.Dump();**

1. Same as #13 only list the books in alphabetical order by title. (9 rows)

**select book\_code, book\_title**

**from books**

**where book\_type in ('sfi', 'mys', 'art')**

**order by book\_title asc;**

**var query14 = from b in Books**

**orderby b.BOOK\_TITLE, b.BOOK\_TYPE**

**where b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “ART”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE };**

**query14.Dump();**

**var query14 = Books.Where(b => b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “ART”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE**

**})**

**.OrderBy(b => b.BOOK\_TITLE);**

**query14.Dump();**

1. Same as #13 but also include the price and list the books in descending order by price. Within a group of books having the same price, further order by book title. (9 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where book\_type in ('sfi', 'mys', 'art')**

**order by book\_price desc, book\_title asc;**

**var query15 = from b in Books**

**orderby b.BOOK\_PRICE**

**where b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “ART”**

**select new { b.BOOK\_CODE, b.BOOK\_TITLE, b.BOOK\_TYPE, b.BOOK\_PRICE };**

**query15.Dump();**

**var query15 = Books.Where(b => b.BOOK\_TYPE == “SFI” ||**

**b.BOOK\_TYPE == “MYS” ||**

**b.BOOK\_TYPE == “ART”)**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE,**

**b.BOOK\_PRICE**

**})**

**.OrderByDescending(b => b.BOOK\_PRICE);**

**query15.Dump();**

1. Display the list of book types in the database. List each book type only once. (12 rows)

**select distinct book\_type**

**from books;**

**var query16 = (from b in Books**

**select new b.BOOK\_TYPE).Distinct();**

**query16.Dump();**

**var query16 = Books.Select(b => b.BOOK\_TYPE)**

**.Distinct();**

**query16.Dump();**

1. How many books have the type SFI? (write sql to produce the answer of 3)

**select count(\*)**

**from books**

**where book\_type = 'sfi';**

**select count(\*)**

**from books**

**where book\_type in ('sfi');**

**var query17 = (from b in Books**

**where b.BOOK\_TYPE == “SFI”**

**select b).Count();**

**query17.Dump();**

**var query17 = Books.Count(b.BOOK\_TYPE == “SFI”)**

**query17.Dump();**

1. For each type of book, list the type and the average price. (Research: GROUP BY - 12 rows)

**select book\_type, avg(book\_price) as average\_price**

**from books**

**group by book\_type;**

**var query18 = from b in Books**

**group b by b.BOOK\_TYPE into g**

**select new { Type = g.Key,**

**Average\_Price = g.Average(b => b.BOOK\_PRICE) };**

**query18.Dump();**

**var query18 = Books.GroupBy(b => b.BOOK\_TYPE)**

**.Select(g => new**

**{**

**Type = g.Key,**

**AveragePrice = g.Average(b => b.BOOK\_PRICE**

**query18.Dump();**

1. Same as #18, but consider only paperback books. (10 rows)

**select book\_type, avg(book\_price) as average\_price**

**from books**

**where book\_paperback = 'Y'**

**group by book\_type;**

**var query19 = from b in Books**

**where b.BOOK\_PAPERBACK== “Y”**

**group b by b.BOOK\_TYPE into g**

**select new { Type = g.Key,**

**Average\_Price = g.Average(b => b.BOOK\_PRICE) };**

**query19.Dump();**

**var query19 = Books.Where(b => b.BOOK\_PAPERBACK == “Y”)**

**.GroupBy(b => b.BOOK\_TYPE)**

**.Select(g => new**

**{**

**Type = g.Key,**

**AveragePrice = g.Average(b => b.BOOK\_PRICE**

**});**

**query19.Dump();**

1. Same as #18, but consider only paperback books for those types which the average price is more than $10. (Research: HAVING. 3 rows)

**select book\_type, avg(book\_price) as average\_price**

**from books**

**where book\_paperback = 'Y'**

**group by book\_type**

**having avg(book\_price) > 10;**

**var query20 = from b in Books**

**where b.BOOK\_PAPERBACK== “Y”**

**group b by b.BOOK\_TYPE into g**

**where g.Average(b => b.BOOK\_PRICE) > 10**

**select new { Type = g.Key,**

**Average\_Price = g.Average(b => b.BOOK\_PRICE) };**

**Query20.Dump();**

**var query20 = Books.Where(b => b.BOOK\_PAPERBACK == “Y”)**

**.GroupBy(b => b.BOOK\_TYPE)**

**.Where(g => g.Average(b => b.BOOK\_PRICE) > 10)**

**.Select(g => new**

**{**

**Type = g.Key,**

**AveragePrice = g.Average(b => b.BOOK\_PRICE**

**});**

**query20.Dump();**

1. What is the most expensive book in the database? (Nested Select - 1 row)

**select max(book\_price) from books;**

**select book\_title from books where book\_price = 37.95;**

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price = (select max(book\_price) from books);**

**var query21 = Books.OrderByDescending(b => b.BOOK\_PRICE).FirstOrDefault();**

**Query21.Dump();**

**var query21 = Books.OrderByDescending(b => b.BOOK\_PRICE)**

**.FirstOrDefault();**

**Query21.Dump();**

1. What are the title(s)/price(s) of the least expensive book(s) in the DB? (3 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price = (select min(book\_price) from books);**

**var query22 = Books.OrderBy (b => b.BOOK\_PRICE).Take(3);**

**Query22.Dump();**

**var query22 = Books.OrderBy(b => b.BOOK\_PRICE)**

**.Take(3);**

**query22.Dump();**

1. How many employees does Henry Books have?

**select sum(branch\_emps) as total\_employees**

**from branches;**

**var query23 = Branches.Where(b.BRANCH\_NAME.StartsWith(“Henry”))**

**.Sum(b => b.BRANCH\_EMPS);**

**Query23.Dump();**

**var query23 = Branches.Sum(b => b.BRANCH\_EMPS);**

**query23.Dump();**

**Console.WriteLine($”There are {query} total employees at Henry Books”);**

**Multiple Table**

1. For each book, list the book code, book title, publisher code, and the publisher name. Order the results by publisher name. (33 rows)

**select b.book\_code, b.book\_title, b.publisher\_code, p.publisher\_name**

**from books b**

**join publishers p**

**on b.publisher\_code = p.publisher\_code;**

**var query1 = Books.Join(Publishers,**

**b => b.PUBLISHER\_CODE,**

**p.PUBLISHER\_CODE, (b, p)**

**=> new {**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.PUBLISHER\_CODE,**

**p.PUBLISHER\_NAME**

**})**

**.OrderBy (p => p.PUBLISHER\_CODE);**

**Query1.Dump();**

**var query1 = from b in books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**orderby p.PUBLISHER\_CODE**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE**

**b.PUBLISHER\_CODE,**

**p.PUBLISHER\_NAME**

**}**

**query1.Dump();**

1. For each book published by Plume, list the book code, book title, and price. (3 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where publisher\_code = (select publisher\_code from publishers where publisher\_name = 'Plume');**

**var query2 = Books.Where(b => Publisher.Any(**

**p => p.PUBLISHER\_CODE == b.PUBLISHER\_CODE**

**&& p.PUBLISHER\_NAME == “Plume”))**

**.Select(b => new {**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**b.PUBLISHER\_CODE**

**});**

**Query2.Dump();**

**var query2 = from b in Books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**where p.PUBLISHER\_NAME == “Plume”**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**b.PUBLISHER\_CODE**

**};**

**query2.Dump();**

1. List the book title, book code, and price of each book published by Plume that has a book price of at least $14. (1 row)

**select book\_code, book\_title, book\_price**

**from books**

**where publisher\_code = (select publisher\_code from publishers where publisher\_name = 'Plume')**

**and book\_price >= 14;**

**var query3 = from b in Books**

**where (b.PUBLISHER\_CODE == “PL” &&**

**b.BOOK\_PRICE >= 14m)**

**select new {**

**b.PUBLISHER\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_CODE,**

**b.BOOK\_PRICE**

**};**

**Query3.Dump();**

**var query3 = from b in Books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**where p.PUBLISHER\_NAME == “Plume” &&**

**b.BOOK\_PRICE >= 14m**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**b.PUBLISHER\_CODE**

**};**

**query3.Dump();**

1. List the book code, book title, and units on hand for each book in branch number 4. (9 rows)

**select b.book\_code, b.book\_title, i.on\_hand**

**from books b**

**join inventory i**

**on b.book\_code = i.book\_code**

**where branch\_number = 4;**

**var query4 = Inventories.Join(Books,**

**i => i.BOOK\_CODE, b => b.BOOK\_CODE**

**(i, b) => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**i.BRANCH\_NUMBER,**

**i.ON\_HAND**

**})**

**.Where (i => i.BRANCH\_NUMBER == 4);**

**Query4.Dump();**

**var query4 = from i in Inventories**

**join b in books on i.BOOK\_CODE equals b.BOOK\_CODE**

**where i.BRANCH\_NUMBER ==4**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**i.ON\_HAND,**

**i.BRANCH\_NUMBER**

**};**

**query4.Dump();**

1. List the book title for each book that has the type PSY and that is published by Jove Publications. (1 row)

**select book\_title**

**from books**

**where book\_type = 'PSY'**

**and publisher\_code = (select publisher\_code from publishers where publisher\_name = 'Jove Publications');**

**var query5 = from b in Books**

**where (b.BOOK\_TYPE == “PSY” &&**

**b.PUBLISHER\_CODE == “JP”)**

**select new {**

**b.BOOK\_TITLE,**

**b.BOOK\_TYPE,**

**b.PUBLISHER\_CODE**

**};**

**Query5.Dump();**

**var query5 = from b in Books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**where b.BOOK\_TYPE == “PSY” &&**

**p.PUBLISHER\_NAME == “Jove Publications”**

**select b.BOOK\_TITLE;**

**query5.Dump();**

1. Find the book title for each book written by author number 18. Use the IN operator in your formulation. (2 rows)

**select b.book\_title**

**from books b**

**join wrote w**

**on b.book\_code = w.book\_code**

**where w.author\_number in (18);**

**var query6 = Wrotes.Where(w => w.AUTHOR\_NUMBER == 18)**

**.Select(w => w.BOOK\_CODE)**

**.Join(Books, wc => wc, b => b.BOOK\_CODE,**

**(wc, b) => b.BOOK\_TITLE);**

**Query6.Dump();**

**var query6 = from b in Books**

**join w in Wrotes on b.BOOK-CODE equals w.BOOK\_CODE**

**where w.AUTHOR\_NUMBER == 18**

**select b.BOOK\_TITLE;**

**query6.Dump();**

1. Repeat exercise 6 but use the EXISTS operator. (2 rows)

**select b.book\_title**

**from books b**

**where exists (select 1 from wrote w where w.book\_code = b.book\_code and w.author\_number = 18);**

**var query7 = from b in Books**

**where Wrotes.Any(w => w.BOOK\_CODE == b.BOOK\_CODE**

**&&w.AUTHOR\_NUMBER == 18)**

**Select b.BOOK\_TITLE;**

**query7.Dump();**

1. Find the book codes and book title for each book located in branch number 2 and written by author 20. (1 row)

**select b.book\_code, b.book\_title**

**from books b**

**join inventory i**

**on b.book\_code = i.book\_code**

**where i.Branch\_number = 2**

**and b.book\_code in**

**(select w.book\_code**

**from wrote w**

**where w.author\_number = 20);**

**var query8 = from b in Books**

**join i in Inventories on b.BOOK\_CODE == equals**

**i.BOOK\_CODE**

**where i.BRANCH\_NUMBER == 2 && Wrotes.Any(**

**w => w.BOOK\_CODE =**

**b.BOOK\_CODE &&**

**w.AUTHOR\_NUMBER == 20)**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE**

**};**

**Query8.Dump();**

**var query8 = Books.Join(Inventories.(Where => i.BRANCH\_NUMBER ==2)**

**b => b.BOOK\_CODE, i => i.BOOK\_CODE, (b, i) => b)**

**.Where(b => Wrotes.Any(w => w.AUTHOR\_NUMBER == 20 && w.BOOK\_CODE == b.BOOK\_CODE))**

**.Select(b => new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE**

**});**

**Query8.Dump();**

1. List book codes for each pair of books that have the same price. (on such pair would be book 0200 and book 7559 because the price of both books is $8.00) Sort the results by first book code and then by second book code. (11 rows)

**select b1.book\_code as book\_code1,**

**b2.book\_code as book\_code2**

**from books b1, books b2**

**where b1.book\_code < b2.book\_code**

**and b1.book\_price = b2.book\_price**

**order by b1.book\_code, b2.book\_code;**

**var query9 = from b1 in Books**

**from b2 in Books**

**where b1.BOOK\_PRICE == b2.BOOK\_PRICE &&**

**b1.BOOK\_CODE.ComapreTo(b2.BOOK\_CODE) < 0**

**select new**

**{**

**Book1 = b1.BOOK\_CODE,**

**Book2 = b2.BOOK\_CODE**

**};**

**Query9.Dump();**

**var query9 = Books.SelectMany(b1 => Books.Where(b2 => b1.BOOK\_PRICE && b2.BOOK\_CODE != b1.BOOK\_CODE)**

**.Select(b2 => new**

**{**

**Book1 = b1.BOOK\_CODE,**

**Book2 = b2.BOOK\_CODE**

**})**

**.Distinct();**

**Query9.Dump();**

1. Find the book title, author last name, and units on hand for each book in branch number 4. (10 rows)

**select b.book\_title, a.author\_last, i.on\_hand**

**from books b**

**join wrote w on b.book\_code = w.book\_code**

**join authors a on w.author\_number = a.author\_number**

**join inventory i on b.book\_code = i.book\_code**

**where i.branch\_number = 4;**

**var query10 = from b in Books**

**join w in Wrotes on b.BOOK\_CODE equals w,BOOK\_CODE join i in Inventories on b.BOOK\_CODE equals i.BOOK\_CODE**

**join a in Authors on w.AUTHOR\_NUMBER equals a.AUTHOR\_NUMBER**

**where i.BRANCH\_NUMBER == 4**

**select new {**

**b.BOOK\_TITLE,**

**a.AUTHOR\_LAST,**

**i.ON\_HAND**

**};**

**Query10.Dump();**

**var query10 = from i in Inventories**

**join b in Books on i.BOOK\_CODE equals b.BOOK\_CODE**

**join w in Wrotes on b.BOOK\_CODE equals w.BOOK\_CODE**

**join a in Authors on w.AUTHOR\_NUMBER equals a.AUTHOR\_NUMBER**

**where i.BRANCH\_NUMBER == 4**

**select new**

**{**

**b.BOOK\_TITLE,**

**a.AUTHOR\_LAST,**

**i.ON\_HAND,**

**i.BRANCH\_NUMBER**

**};**

**Query10.Dump();**

1. Repeat exercise 10 but list only paperback books. (4 rows)

**select b.book\_title, a.author\_last, i.on\_hand**

**from books b**

**join wrote w on b.book\_code = w.book\_code**

**join authors a on w.author\_number = a.author\_number**

**join inventory i on b.book\_code = i.book\_code**

**where i.branch\_number = 4**

**and b.book\_paperback = 'Y';**

**var query11 = from b in Books**

**join w in Wrotes on b.BOOK\_CODE equals w,BOOK\_CODE join i in Inventories on b.BOOK\_CODE equals i.BOOK\_CODE**

**join a in Authors on w.AUTHOR\_NUMBER equals a.AUTHOR\_NUMBER**

**where i.BRANCH\_NUMBER == 4 &&**

**b.BOOK\_PAPERBACK == “Y”**

**select new {**

**b.BOOK\_TITLE,**

**a.AUTHOR\_LAST,**

**i.ON\_HAND**

**};**

**Query11.Dump();**

**var query11 = from i in Inventories**

**join b in Books on i.BOOK\_CODE equals b.BOOK\_CODE**

**join w in Wrotes on b.BOOK\_CODE equals w.BOOK\_CODE**

**join a in Authors on w.AUTHOR\_NUMBER equals a.AUTHOR\_NUMBER**

**where i.BRANCH\_NUMBER == 4 &&**

**b.BOOK\_PAPERBACK == “Y”**

**select new**

**{**

**b.BOOK\_TITLE,**

**a.AUTHOR\_LAST,**

**i.ON\_HAND,**

**i.BRANCH\_NUMBER**

**};**

**Query11.Dump();**

1. Find the book code and book title for each book whose price is more than $10 and that was published in Boston. (Research Union keyword. 20 rows)

**select b.book\_code, b.book\_title**

**from books b**

**join publishers p on b.publisher\_code = p.publisher\_code**

**where book\_price > 10**

**and publisher\_city = 'Boston';**

**var query12 = from b in Books**

**join p in Publishers on o.PUBLISHER\_CODE = p.PUBLISHER\_CODE**

**where b.BOOK\_PRICE > 10m && p.PUBLISHER\_CITY == “Boston”**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE**

**};**

**Query12.Dump();**

**var query12 = from b in Books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**orderby b.BOOK\_PRICE**

**where b.BOOK-PRICE > 10 &&**

**p.PUBLISHER\_CITY == “Boston”**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**p.PUBLISHER\_CITY**

**};**

**Query12.Dump();**

1. Find the book code and book title for each book whose price is more than $10 and that was not published in Boston. (18 rows)

**select b.book\_code, b.book\_title**

**from books b**

**join publishers p on b.publisher\_code = p.publisher\_code**

**where book\_price > 10**

**and publisher\_city != 'Boston';**

**var query13 = from b in Books**

**join p in Publishers on o.PUBLISHER\_CODE = p.PUBLISHER\_CODE**

**where b.BOOK\_PRICE > 10m && p.PUBLISHER\_CITY != “Boston”**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE**

**};**

**Query13.Dump();**

**var query13 = from b in Books**

**join p in Publishers on b.PUBLISHER\_CODE equals p.PUBLISHER\_CODE**

**orderby p.PUBLISHER\_CITY, b.BOOK\_PRICE**

**where b.BOOK-PRICE > 10 &&**

**p.PUBLISHER\_CITY != “Boston”**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE,**

**p.PUBLISHER\_CITY**

**};**

**Query13.Dump();**

1. Find the book code and book title for each book whose price is greater than the book price for every book that has the type HOR. (5 rows)

**select book\_code, book\_title, book\_price**

**from books**

**where book\_price > all**

**(select book\_price**

**from books**

**where book\_type = 'HOR')**

**var query14 = fromb1 in Books**

**where b1.BOOK\_PRICE > (from b2 in Books where**

**b2.BOOK\_PRICE == “HOR”**

**select b2.BOOK\_PRICE).MAX()**

**select new**

**{**

**b1.BOOK\_CODE**

**b1.BOOK\_TITLE**

**b1.BOOK\_ PRICE**

**};**

**Query14.Dump();**

**var query14 = from b in Books**

**where b.BOOK-PRICE > (from b2 in Books where b2.BOOK\_TYPE == “HOR”**

**select b.BOOK\_PRICE).Max()**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE**

**};**

**Query14.Dump();**

1. Find the book code and book title for each book whose price is greater than the price of at least one book that has the type HOR. (23 rows)

**select book\_code, book\_title**

**from books**

**where book\_price > any**

**(select book\_price**

**from books**

**where book\_type = 'HOR')**

**order by book\_price;**

**var query15 = from b1 in Books**

**orderby b1.BOOK\_PRICE**

**where b1.BOOK\_PRICE > (from b2 in Books where**

**select b2.BOOK\_PRICE == “HOR”**

**b2.BOOK\_PRICE).FirstOrDefault()**

**select new**

**{**

**b1.BOOK\_CODE**

**b1.BOOK\_TITLE**

**b1.BOOK\_ PRICE**

**};**

**Query15.Dump();**

**var query15 = from b in Books**

**where b.BOOK\_PRICE > (from b2 in Books where b2.BOOK\_TYPE == “HOR”**

**select b2.BOOK\_PRICE).FirstOrDefault()**

**orderby b.BOOK\_PRICE**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**b.BOOK\_PRICE**

**};**

**Query15.Dump();**

1. List the book code, book title, and units on hand for each book in branch number 2. Be sure each book is included, regardless of whether there are copies of the book currently on hand in branch 2. Order the output by book code. (Research Left Join: 33 rows)

**select b.book\_code, b.book\_title, coalesce(i.on\_hand, 0) as onhand**

**from books b**

**left join inventory i on b.book\_code = i.book\_code**

**and i.branch\_number = 2**

**order by b.book\_code;**

**var query16 = from b in Books**

**join inventory in Inventories on**

**b.BOOK\_CODE equals inventory.BOOK\_CODE into**

**invGroup**

**from inv in invGroup.DefaultIfEmpty()**

**where inv.BRANCH\_NUMBER == 2 || inv == null**

**orderby b.BOOK\_CODE**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**ON\_HAND = inv != null ? inv.ON\_HAND : 0**

**Query16.Dump();**

**var query16 = from b in Books**

**join i in Inventories on b.BOOK\_CODE equals i.BOOK\_CODE into invGroup**

**from inv in invGroup.DefaultIfEmpty()**

**where inv.BRANCH\_NUMBER == 2 || inv == null**

**orderby b.BOOK\_CODE**

**select new**

**{**

**b.BOOK\_CODE,**

**b.BOOK\_TITLE,**

**ON\_HAND = inv != null ? inv.ON\_HAND : 0**

**};**

**Query16.Dump();**