

```
1 DROP DATABASE IF EXISTS Jungle_Book;
2
3 CREATE DATABASE Jungle_Book;
4
5 USE Jungle_Book;
6
7 -- CREATE TABLES
8 -- creating the customer table
9 CREATE TABLE Customer
10 (
11     CustomerID INT(11) NOT NULL AUTO_INCREMENT,
12     CustomerFirstName VARCHAR(32) NOT NULL,
13     CustomerLastName VARCHAR(32) NOT NULL,
14     BillingAddress VARCHAR(255) NOT NULL,
15     ShippingAddress VARCHAR(255) NOT NULL,
16     PhoneNum VARCHAR(11) NOT NULL,
17     EmailAddress VARCHAR(80) NOT NULL,
18     Username VARCHAR(32) NOT NULL,
19     Password VARCHAR(32) NOT NULL,
20     PRIMARY KEY (CustomerID)
21 );
22
23 -- creating the book table
24 CREATE TABLE Book
25 (
26     ISBN VARCHAR(13) NOT NULL,
27     BookTitle VARCHAR(80) NOT NULL,
28     BriefSummary VARCHAR(255) NOT NULL,
29     Price FLOAT NOT NULL,
30     PRIMARY KEY (ISBN)
31 );
32
33 -- creating author table
34 CREATE TABLE Author
35 (
36     AuthorID INT(11) NOT NULL AUTO_INCREMENT,
37     AuthorFirstName VARCHAR(32) NOT NULL,
38     AuthorLastName VARCHAR(32) NOT NULL,
39     PRIMARY KEY (AuthorID)
40 );
41
42 -- create order table
43 CREATE TABLE CustomerOrder
44 (
45     OrderNum INT(11) NOT NULL AUTO_INCREMENT,
46     DateOrdered DATETIME NOT NULL,
```

```

47     DateShipped DATETIME NULL,
48     OrderStatus VARCHAR(32) NOT NULL,
49     OrderSubtotal FLOAT NOT NULL,
50     Taxes FLOAT NOT NULL,
51     TotalCost FLOAT NOT NULL,
52     CustomerID INT(11) NOT NULL,
53     PRIMARY KEY (OrderNum)
54 );
55
56 -- create book category
57 CREATE TABLE BookCategory
58 (
59     BookCategoryID INT(11) NOT NULL AUTO_INCREMENT,
60     BookCategoryName VARCHAR(32) NOT NULL,
61     PRIMARY KEY (BookCategoryID)
62 );
63
64 -- create OrderBook
65 CREATE TABLE Order_Book
66 (
67     OrderNum INT(11) NOT NULL,
68     ISBN VARCHAR(13) NOT NULL,
69     Quantity INT(11) NOT NULL,
70     Subtotal FLOAT NOT NULL,
71     PRIMARY KEY (OrderNum, ISBN)
72 );
73
74 -- create Author_Book
75 CREATE TABLE Author_Book
76 (
77     AuthorID INT(11) NOT NULL,
78     ISBN VARCHAR(13) NOT NULL,
79     PRIMARY KEY (AuthorID, ISBN)
80 );
81
82 -- create Book_BookCategory
83 CREATE TABLE Book_BookCategory
84 (
85     ISBN VARCHAR(13) NOT NULL,
86     BookCategoryID INT(11) NOT NULL,
87     PRIMARY KEY (ISBN, BookCategoryID)
88 );
89
90 -- ALTERING TABLES
91 ALTER TABLE CustomerOrder
92     ADD CONSTRAINT FK_CustomerOrder_CustomerID

```

```
93     FOREIGN KEY (CustomerID)
94     REFERENCES Customer (CustomerID);
95
96 ALTER TABLE Author_Book
97     ADD CONSTRAINT FK_Author_Book_AuthorID
98     FOREIGN KEY (AuthorID)
99     REFERENCES Author (AuthorID);
100
101 ALTER TABLE Author_Book
102     ADD CONSTRAINT FK_Author_Book_ISBN
103     FOREIGN KEY (ISBN)
104     REFERENCES Book (ISBN);
105
106 ALTER TABLE Order_Book
107     ADD CONSTRAINT FK_Order_Book_OrderNum
108     FOREIGN KEY (OrderNum)
109     REFERENCES CustomerOrder (OrderNum);
110
111 ALTER TABLE Order_Book
112     ADD CONSTRAINT FK_Order_Book_ISBN
113     FOREIGN KEY (ISBN)
114     REFERENCES Book (ISBN);
115
116 ALTER TABLE Book_BookCategory
117     ADD CONSTRAINT FK_Book_BookCategory_ISBN
118     FOREIGN KEY (ISBN)
119     REFERENCES Book (ISBN);
120
121 ALTER TABLE Book_BookCategory
122     ADD CONSTRAINT FK_Book_BookCategory_BookCategoryID
123     FOREIGN KEY (BookCategoryID)
124     REFERENCES BookCategory (BookCategoryID);
125
126 -- adding the categories to Book Category
127 INSERT INTO BookCategory
128     (BookCategoryName)
129 VALUES
130     ('Fantasy');
131
132 INSERT INTO BookCategory
133     (BookCategoryName)
134 VALUES
135     ('Mystery');
136
137 INSERT INTO BookCategory
138     (BookCategoryName)
```

```
139 VALUES
140     ('Horror');
141
142 INSERT INTO BookCategory
143     (BookCategoryName)
144 VALUES
145     ('Thriller');
146
147 INSERT INTO BookCategory
148     (BookCategoryName)
149 VALUES
150     ('Databases');
151
152 -- adding books to Book
153 INSERT INTO Book
154     (ISBN, BookTitle, BriefSummary, Price)
155 VALUES
156     ('9780133970777', 'Fundamentals of Database Systems (
157         'This book introduces the fundamental concepts
158         necessary for designing,
159         using, and implementing database systems and
160         database applications. ' , 179.16);
161
162 INSERT INTO Book
163     (ISBN, BookTitle, BriefSummary, Price)
164 VALUES
165     ('9780307269751', 'The Girl with the Dragon Tattoo',
166         'An exciting, stunningly intelligent novel about
167         society at its most hidden,
168         and about the intimate lives of a brilliantly
169         realized cast of characters,
170         all of them forced to face the darker aspects of
171         their world and of their own lives.' , 35.00);
172
173 INSERT INTO Book
174     (ISBN, BookTitle, BriefSummary, Price)
175 VALUES
176     ('9780385121675', 'The Shining',
177         'Jack Torrance gets a new job at the hotel and spends
178         time with his family and his writing.
179         Winter comes, the hotel feels remote and sinister,
180         the only one to notice the strange and terrible
181         forces is Danny Torrance, a unique 5 year old.' , 49.99);
182
183 -- adding authors to Author
```

```
177 INSERT INTO Author
178     (AuthorFirstName, AuthorLastName)
179 VALUES
180     ('Ramez', 'Elmasri');
181
182 INSERT INTO Author
183     (AuthorFirstName, AuthorLastName)
184 VALUES
185     ('Shamkant', 'Navathe');
186
187 INSERT INTO Author
188     (AuthorFirstName, AuthorLastName)
189 VALUES
190     ('Stephen', 'King');
191
192 INSERT INTO Author
193     (AuthorFirstName, AuthorLastName)
194 VALUES
195     ('Stieg', 'Larsson');
196
197 -- adding the author to each book
198 INSERT INTO Author_Book
199     (AuthorID, ISBN)
200 VALUES
201     (4, 9780307269751);
202
203 INSERT INTO Author_Book
204     (AuthorID, ISBN)
205 VALUES
206     (3, 9780385121675);
207
208 INSERT INTO Author_Book
209     (AuthorID, ISBN)
210 VALUES
211     (2, 9780133970777);
212
213 INSERT INTO Author_Book
214     (AuthorID, ISBN)
215 VALUES
216     (1, 9780133970777);
217
218 -- adding categories to each book
219 INSERT INTO Book_BookCategory
220     (ISBN, BookCategoryID)
221 VALUES
222     ('9780307269751', '2');
```

```
223
224 INSERT INTO Book_BookCategory
225     (ISBN, BookCategoryID)
226 VALUES
227     ('9780307269751', '4');
228
229 INSERT INTO Book_BookCategory
230     (ISBN, BookCategoryID)
231 VALUES
232     ('9780385121675', '2');
233
234 INSERT INTO Book_BookCategory
235     (ISBN, BookCategoryID)
236 VALUES
237     ('9780385121675', '3');
238
239 INSERT INTO Book_BookCategory
240     (ISBN, BookCategoryID)
241 VALUES
242     ('9780133970777', '5');
243
244 -- adding customers into Customer
245 INSERT INTO Customer
246     (CustomerFirstName, CustomerLastName, BillingAddress,
247      ShippingAddress, PhoneNum, EmailAddress, Username,
248      Password)
249 VALUES
250     ('Monica', 'Geller',
251      '77 Somewhere Lane', '77 Somewhere Lane',
252      '9052177477', 'monica.geller@gmail.com',
253      'monCleanFreak', 'monica77');
254
255 INSERT INTO Customer
256     (CustomerFirstName, CustomerLastName, BillingAddress,
257      ShippingAddress, PhoneNum, EmailAddress, Username,
258      Password)
259 VALUES
260     ('Chandler', 'Bing',
261      '15 Area Road', '90 Anywhere Avenue',
262      '4754801478', 'chandler.bing@yahoo.com',
263      'missChanandelerBong', 'bing123');
264
265 INSERT INTO Customer
266     (CustomerFirstName, CustomerLastName, BillingAddress,
267      ShippingAddress, PhoneNum, EmailAddress, Username,
268      Password)
```

```

263 VALUES
264     ('Rachel', 'Green',
265      '109 Fashion Avenue', '109 Fashion Avenue',
266      '5559287541', 'rachel_green@hotmail.com',
267      'greenMachine', 'emma234');
268
269 INSERT INTO Customer
270     (CustomerFirstName, CustomerLastName, BillingAddress,
271      ShippingAddress, PhoneNum, EmailAddress, Username,
272      Password)
273 VALUES
274     ('Joseph', 'Tribbiani',
275      '44 Unknown Street', '15 Area Road',
276      '8254457859', 'joey_tribbiani@icloud.com',
277      'joeyTrib', 'joey22');
278
279 -- creating 3 new orders
280 -- customer one buys one copy of all three books
281 INSERT INTO CustomerOrder
282     (DateOrdered, DateShipped, OrderStatus, OrderSubtotal
283      , Taxes, TotalCost, CustomerID)
284 VALUES
285     ('2019-11-23T09:07:17.000', NULL, 'pending
286      confirmation', 264.15, 39.62, 303.77, 1);
287 -- adding the items in the customer's order
288 -- customer 1, order 1
289 INSERT INTO Order_Book
290     (OrderNum, ISBN, Quantity, Subtotal)
291 VALUES
292     (1, '9780307269751', 1, 35);
293
294 INSERT INTO Order_Book
295     (OrderNum, ISBN, Quantity, Subtotal)
296 VALUES
297     (1, '9780385121675', 1, 49.99);
298
299 INSERT INTO Order_Book
300     (OrderNum, ISBN, Quantity, Subtotal)
301 VALUES
302     (1, '9780133970777', 1, 179.16);
303
304 -- customer 2 buys just one of the books
305 INSERT INTO CustomerOrder
306     (DateOrdered, DateShipped, OrderStatus, OrderSubtotal

```

```

304 , Taxes, TotalCost, CustomerID)
305 VALUES
306     ('2019-11-22T10:04:02.000', NULL, 'pending
confirmation', 49.99, 7.50, 57.49, 2);
307
308 -- adding the items in the customer's order
309 INSERT INTO Order_Book
310     (OrderNum, ISBN, Quantity, Subtotal)
311 VALUES
312     (2, '9780385121675', 1, 49.99);
313
314
315 -- customer 3 buys five copies of one book, two copies of
the second book, and ten copies of the third
316 INSERT INTO CustomerOrder
317     (DateOrdered, DateShipped, OrderStatus, OrderSubtotal
, Taxes, TotalCost, CustomerID)
318 VALUES
319     ('2019-11-25T05:12:02.000', NULL, 'pending
confirmation', 1465.70, 73.29, 1685.56, 3);
320
321 -- adding the item's in the customer's order
322 INSERT INTO Order_Book
323     (OrderNum, ISBN, Quantity, Subtotal)
324 VALUES
325     (3, '9780133970777', 5, 895.80);
326
327 INSERT INTO Order_Book
328     (OrderNum, ISBN, Quantity, Subtotal)
329 VALUES
330     (3, '9780307269751', 2, 70);
331
332 INSERT INTO Order_Book
333     (OrderNum, ISBN, Quantity, Subtotal)
334 VALUES
335     (3, '9780385121675', 10, 499.90);
336
337 -- delete the second order
338 DELETE FROM Order_Book
339     WHERE OrderNum = 2;
340 DELETE FROM CustomerOrder
341     WHERE CustomerID = 2;
342
343 -- change the status of the first order to complete
344 UPDATE CustomerOrder
345 SET DateShipped = '2019-11-26T04:12:02.000', OrderStatus

```



```

345 = 'complete'
346 WHERE CustomerID = 1 AND OrderNum = 1;
347
348 -- change the third order to add another copy of the
    second book
349 UPDATE Order_Book
350 SET Quantity = 3, Subtotal = 105
351 WHERE OrderNum = 3 AND ISBN = '9780307269751';
352 UPDATE CustomerOrder
353 SET OrderSubtotal = 1500.70, Taxes = 75.04, TotalCost =
    1725.81
354 WHERE CustomerID = 3 AND OrderNum = 3;
355
356 -- display all customer information for customers with no
    orders
357 SELECT C.CustomerID, CustomerFirstName, CustomerLastName,
    BillingAddress, ShippingAddress, PhoneNum, EmailAddress,
    Username, Password
358 FROM Customer C LEFT OUTER JOIN CustomerOrder CO
359 ON C.CustomerID = CO.CustomerID
360 WHERE OrderNum is NULL;
361
362 -- display the title, author, ISBN, and price of all books
363 SELECT B.ISBN, BookTitle, Price, A.AuthorID,
    AuthorLastName, AuthorFirstName
364 FROM Book B INNER JOIN Author_Book AB ON B.ISBN = AB.ISBN
365         INNER JOIN Author A ON AB.AuthorID = A.
    AuthorID
366         INNER JOIN Book_BookCategory BBC ON AB.ISBN =
    BBC.ISBN
367         INNER JOIN BookCategory BC ON BBC.
    BookCategoryID = BC.BookCategoryID
368 WHERE BookTitle LIKE '%Database%' OR
    BookCategoryName LIKE '%Database%';
369
370 -- display the email addresses of customers who have
    outstanding orders
371 SELECT EmailAddress
372 FROM Customer C RIGHT OUTER JOIN CustomerOrder CO
373 ON C.CustomerID = CO.CustomerID
374 WHERE DateShipped is NULL;
375
376 -- display just the order information on all orders that
    have purchased more than one copy of any books
377 SELECT CustomerOrder.OrderNum, CustomerOrder.DateOrdered,
    CustomerOrder.DateShipped, CustomerOrder.OrderStatus,

```

```
378         CustomerOrder.OrderSubtotal, CustomerOrder.Taxes,
        CustomerOrder.TotalCost, CustomerOrder.CustomerID
379 FROM CustomerOrder RIGHT OUTER JOIN Order_Book OB
380 ON CustomerOrder.OrderNum = OB.OrderNum
381 WHERE Quantity > 1
382 GROUP BY OB.OrderNum
383 ORDER BY OB.OrderNum;
384
385 -- display the order number and total number of books in
    each order
386 SELECT CustomerOrder.OrderNum, SUM(Quantity)
387 FROM CustomerOrder
388 INNER JOIN Order_Book OB
389 ON CustomerOrder.OrderNum = OB.OrderNum
390 GROUP BY CustomerOrder.OrderNum;
391
392
393 -- display each order number, customer name, and total
    cost of the order
394 SELECT CO.OrderNum, CustomerLastName, CustomerFirstName,
    SUM(Quantity * Price * 1.15)
395 FROM CustomerOrder CO INNER JOIN Customer C ON CO.
    CustomerID = C.CustomerID
396         INNER JOIN Order_Book OB ON CO.
    OrderNum = OB.OrderNum
397         INNER JOIN Book B ON OB.ISBN = B.
    ISBN
398 GROUP BY CO.OrderNum;
399
400
```