```sql
1
2  -- create the database
3  DROP DATABASE IF EXISTS WestConCableCompany;
4  CREATE DATABASE WestConCableCompany;
5
6  USE WestConCableCompany;
7
8
9  -- create the tables
10 DROP TABLE IF EXISTS Customer;
11 CREATE TABLE Customer (
12     CustomerID INT PRIMARY KEY,
13     CustomerName VARCHAR(32) NOT NULL,
14     Payment FLOAT NULL
15 );
16
17 DROP TABLE IF EXISTS Channel;
18 CREATE TABLE Channel(
19     ChannelID INT PRIMARY KEY,
20     ChannelName VARCHAR(32) NOT NULL
21 );
22
23 DROP TABLE IF EXISTS CablePackage;
24 CREATE TABLE CablePackage(
25     CablePackageID INT PRIMARY KEY,
26     CablePackageName VARCHAR(32) NOT NULL
27 );
28
29 DROP TABLE IF EXISTS PayPerView;
30 CREATE TABLE PayPerView(
31     PayPerViewID INT PRIMARY KEY,
32     PayPerViewName VARCHAR(32) NOT NULL,
33     EventDate DATETIME NOT NULL,
34     Price FLOAT NOT NULL,
35     ChannelID INT NOT NULL
36 );
37
38 DROP TABLE IF EXISTS CablePackageChannel;
39 CREATE TABLE CablePackageChannel(
40     CablePackageID INT NOT NULL,
41     ChannelID INT NOT NULL
42 );
43
44 DROP TABLE IF EXISTS CustomerPayPerView;
45 CREATE TABLE CustomerPayPerView(
```

```sql
46      CustomerID INT NOT NULL,
47      PayPerViewID INT NOT NULL
48 );
49
50 DROP TABLE IF EXISTS CustomerCablePackage;
51 CREATE TABLE CustomerCablePackage(
52      CustomerID INT NOT NULL,
53      CablePackageID INT NOT NULL
54 );
55
56
57 -- create relationships
58 ALTER TABLE PayPerView
59      DROP CONSTRAINT IF EXISTS FK__Channel__ChannelID;
60 ALTER TABLE PayPerView
61      ADD CONSTRAINT FK__Channel__ChannelID FOREIGN KEY (
   ChannelID)
62          REFERENCES Channel (ChannelID);
63
64
65 ALTER TABLE CablePackageChannel
66      DROP CONSTRAINT IF EXISTS PK_CablePackageChannel;
67 ALTER TABLE CablePackageChannel
68      ADD CONSTRAINT  PK_CablePackageChannel
69      PRIMARY KEY (CablePackageID, ChannelID);
70
71 ALTER TABLE CablePackageChannel
72      DROP CONSTRAINT IF EXISTS
   FK_CablePackage_CablePackageID;
73 ALTER TABLE CablePackageChannel
74      ADD CONSTRAINT  FK_CablePackage_CablePackageID
75      FOREIGN KEY (CablePackageID)
76      REFERENCES CablePackage (CablePackageID);
77
78 ALTER TABLE CablePackageChannel
79      DROP CONSTRAINT IF EXISTS FK_Channel_ChannelID;
80 ALTER TABLE CablePackageChannel
81      ADD CONSTRAINT  FK_Channel_ChannelID
82      FOREIGN KEY (ChannelID)
83      REFERENCES Channel (ChannelID);
84
85
86 ALTER TABLE CustomerPayPerView
87      DROP CONSTRAINT IF EXISTS PK_CustomerPayPerView;
88 ALTER TABLE CustomerPayPerView
```

```sql
 89        ADD CONSTRAINT  PK_CustomerPayPerView
 90        PRIMARY KEY (CustomerID, PayPerViewID);
 91
 92  ALTER TABLE CustomerPayPerView
 93        DROP CONSTRAINT IF EXISTS FK_Customer_CustomerID;
 94  ALTER TABLE CustomerPayPerView
 95        ADD CONSTRAINT  FK_Customer_CustomerID
 96        FOREIGN KEY (CustomerID)
 97        REFERENCES Customer (CustomerID);
 98
 99  ALTER TABLE CustomerPayPerView
100        DROP CONSTRAINT IF EXISTS FK_PayPerView_PayPerViewID;
101  ALTER TABLE CustomerPayPerView
102        ADD CONSTRAINT  FK_PayPerView_PayPerViewID
103        FOREIGN KEY (PayPerViewID)
104        REFERENCES PayPerView (PayPerViewID);
105
106
107  ALTER TABLE CustomerCablePackage
108        DROP CONSTRAINT IF EXISTS PK_CustomerCablePackage;
109  ALTER TABLE CustomerCablePackage
110        ADD CONSTRAINT  PK_CustomerCablePackage
111        PRIMARY KEY (CustomerID, CablePackageID);
112
113  ALTER TABLE CustomerCablePackage
114        DROP CONSTRAINT IF EXISTS FK_CustomerC_CustomerID;
115  ALTER TABLE CustomerCablePackage
116        ADD CONSTRAINT  FK_CustomerC_CustomerID
117        FOREIGN KEY (CustomerID)
118        REFERENCES Customer (CustomerID);
119
120  ALTER TABLE CustomerCablePackage
121        DROP CONSTRAINT IF EXISTS
     FK_CablePackageC_CablePackageID;
122  ALTER TABLE CustomerCablePackage
123        ADD CONSTRAINT  FK_CablePackageC_CablePackageID
124        FOREIGN KEY (CablePackageID)
125        REFERENCES CablePackage (CablePackageID);
126
127
128  -- INSERTing customers
129  INSERT INTO Customer
130        (CustomerID, CustomerName)
131  VALUES
132        (1, 'John Buck'), (2, 'Jane Doe'), (3, 'Jack Sprat');
```

```sql
133
134
135  -- INSERTing channels
136  INSERT INTO Channel
137      (ChannelID, ChannelName)
138  VALUES
139      (1, 'ABC (100)'), (2, 'Spike (106)'), (3, 'TSN (110)'
         );
140
141
142  -- INSERTing cable packages
143  INSERT INTO CablePackage
144      (CablePackageID, CablePackageName)
145  VALUES
146      (1, 'Basic'), (2, 'Action'), (3, 'Sports');
147
148
149  -- INSERTing channels in cable packages
150  INSERT INTO CablePackageChannel
151      (CablePackageID, ChannelID)
152  VALUES
153      (1, 1), (2, 2), (3, 3);
154
155
156  -- INSERTing Basic package to all customers
157  INSERT INTO CustomerCablePackage
158      (CablePackageID, CustomerID)
159  VALUES
160      (1, 1), (1, 2), (1, 3);
161
162
163  -- INSERTing Action package to Jane Doe
164  INSERT INTO CustomerCablePackage
165      (CablePackageID, CustomerID)
166  VALUES
167      (2, 2);
168
169
170  -- INSERTing pay per view
171  INSERT INTO PayPerView
172      (PayPerViewID, PayPerViewName, EventDate, Price,
         ChannelID)
173  VALUES
174      (2, 'The Prize Fight', '2018-10-31T20:00:000', 9.95,
         3);
```

```sql
175
176
177  -- INSERTing John Buck to PayPerView The Prize Fight
178  INSERT INTO CustomerPayPerView
179      (CustomerID, PayPerViewID)
180  VALUES
181      (1, 1);
182
183
184  -- UPDATEing John Buck's payment
185  UPDATE Customer
186  SET Payment = 9.95
187  WHERE CustomerID = 1;
188
189
190  -- DELETEing Jane Doe's cable package
191  DELETE FROM CustomerCablePackage
192  WHERE CustomerID = 2 AND CablePackageID = 2;
193
194
195  -- SELECTing all customer information alphabetically by
     last name
196  SELECT * FROM Customer;
197  ORDER BY SUBSTR(CustomerName, INSTR(CustomerName, ' '));
198
199
200  -- SELECTing all information for the Spike channel
201  SELECT * FROM Channel
202  WHERE ChannelName LIKE 'Spike';
203
204
205  -- SELECTing the title and date of all future Pay-Per-
     View events
206  SELECT PayPerViewName, EventDate FROM PayPerView
207  WHERE EventDate > CURDATE();
```