```sql
 1  -- move to default database
 2  USE master;
 3  GO
 4
 5  -- create a new database
 6  DROP DATABASE IF EXISTS  SpeedingTicket;
 7  CREATE DATABASE SpeedingTicket;
 8  GO
 9
10  -- switch to new database
11  USE SpeedingTicket;
12  GO
13
14  -- CREATE TABLES
15  -- table: User
16  DROP TABLE IF EXISTS [User]
17  CREATE TABLE [User]
18  (
19      UserID INT IDENTITY NOT NULL,
20      UserFirstName VARCHAR(32) NOT NULL,
21      UserLastName VARCHAR(32) NOT NULL,
22      UserAddress  VARCHAR(255) NOT NULL,
23      UserPhoneNum VARCHAR(32) NOT NULL,
24      PRIMARY KEY (UserID)
25  );
26  GO
27
28  -- table: Staff
29  DROP TABLE IF EXISTS Staff;
30  CREATE TABLE Staff (
31      StaffID INT IDENTITY NOT NULL,
32      StaffFirstName VARCHAR(32) NOT NULL,
33      StaffLastName VARCHAR(32) NOT NULL,
34      StaffPhoneNum VARCHAR(32) NOT NULL,
35      StaffAddress VARCHAR(255) NOT NULL,
36      PRIMARY KEY (StaffID)
37  );
38  GO
39
40  -- table: Ticket
41  DROP TABLE IF EXISTS Ticket;
42  CREATE TABLE Ticket (
43      TicketID INT IDENTITY NOT NULL,
44      Issue VARCHAR(1024) NOT NULL,
45      DateIssued DATETIME NOT NULL,
46      DateResolved DATETIME NULL,
```

```sql
47      UserID INT NULL,
48      StaffID INT NULL,
49      TicketCategoryID INT NOT NULL,
50      PRIMARY KEY (TicketID)
51 );
52 GO
53
54 -- table: TicketCategory
55 DROP TABLE IF EXISTS TicketCategory;
56 CREATE TABLE TicketCategory (
57      TicketCategoryID INT IDENTITY NOT NULL,
58      CategoryName VARCHAR(32) NOT NULL,
59      CategoryDescription VARCHAR(255) NOT NULL,
60      PRIMARY KEY (TicketCategoryID)
61 );
62 GO
63
64 -- table: Comment
65 DROP TABLE IF EXISTS Comment;
66 CREATE TABLE Comment (
67      CommentID INT IDENTITY NOT NULL,
68      CommentMade VARCHAR(1024) NOT NULL,
69      TicketID INT NOT NULL,
70      UserID INT NULL,
71      StaffID INT NULL,
72      PRIMARY KEY (CommentID)
73 );
74 GO
75
76 -- table: Task
77 DROP TABLE IF EXISTS Task;
78 CREATE TABLE Task (
79      TaskID INT IDENTITY NOT NULL,
80      TaskName VARCHAR(32) NOT NULL,
81      TaskInformation VARCHAR(255) NOT NULL,
82      TicketID INT NOT NULL,
83      StaffID INT NOT NULL,
84      PRIMARY KEY (TaskID)
85 );
86 GO
87
88 -- table: Hardware
89 DROP TABLE IF EXISTS Hardware;
90 CREATE TABLE Hardware (
91      HardwareID INT IDENTITY NOT NULL,
92      HardwareName VARCHAR(32) NOT NULL,
```

```
 93        HardwareDescription VARCHAR(80) NOT NULL,
 94        PurchaseDate DATETIME NOT NULL,
 95        SerialNumber VARCHAR(32) NOT NULL
 96        PRIMARY KEY (HardwareID)
 97 );
 98 GO
 99
100 -- table: HardwareTicket
101 DROP TABLE IF EXISTS HardwareTicket;
102 CREATE TABLE HardwareTicket (
103        TicketID INT NOT NULL,
104        HardwareID INT NOT NULL,
105        PRIMARY KEY (TicketID, HardwareID)
106 );
107 GO
108
109
110 -- ALTER TABLES
111 -- foreign key constraints
112 -- table: Ticket
113 ALTER TABLE Ticket
114 ADD CONSTRAINT FK__Ticket__UserID
115 FOREIGN KEY (UserID)
116 REFERENCES [User] (UserID) ON DELETE SET NULL ON UPDATE
    SET NULL;
117 GO
118
119 ALTER TABLE Ticket
120 ADD CONSTRAINT FK__Ticket__StaffID
121 FOREIGN KEY (StaffID)
122 REFERENCES Staff (StaffID) ON DELETE SET NULL ON UPDATE
    SET NULL;
123 GO
124
125 ALTER TABLE Ticket
126 WITH CHECK ADD CONSTRAINT FK__Ticket__TicketCategoryID
127 FOREIGN KEY (TicketCategoryID)
128 REFERENCES TicketCategory (TicketCategoryID) ON DELETE
    CASCADE ON UPDATE CASCADE;
129 GO
130
131 -- table: Comment
132 ALTER TABLE Comment
133 ADD CONSTRAINT FK__Comment__TicketID
134 FOREIGN KEY (TicketID)
135 REFERENCES Ticket (TicketID) ON DELETE CASCADE ON UPDATE
```

```
135 CASCADE;
136 GO
137
138 ALTER TABLE Comment
139 ADD CONSTRAINT FK__Comment__UserID
140 FOREIGN KEY (UserID)
141 REFERENCES [User] (UserID);
142 GO
143
144 ALTER TABLE Comment
145 WITH CHECK ADD CONSTRAINT FK__User__UserID
146 FOREIGN KEY (UserID)
147 REFERENCES [User] (UserID);
148 GO
149
150 -- table: Task
151 ALTER TABLE Task
152 ADD CONSTRAINT FK__Task__TicketID
153 FOREIGN KEY (TicketID)
154 REFERENCES Ticket (TicketID) ON DELETE CASCADE ON UPDATE
    CASCADE;
155 GO
156
157 ALTER TABLE Task
158 ADD CONSTRAINT FK__Task__StaffID
159 FOREIGN KEY (StaffID)
160 REFERENCES Staff (StaffID);
161 GO
162
163 -- table: HardwareTicket
164 ALTER TABLE HardwareTicket
165 ADD CONSTRAINT FK__HardwareTicket__HardwareID
166 FOREIGN KEY (TicketID)
167 REFERENCES Ticket (TicketID);
168 GO
169
170 ALTER TABLE HardwareTicket
171 WITH CHECK ADD CONSTRAINT FK__Item__ItemID
172 FOREIGN KEY (HardwareID)
173 REFERENCES Hardware (HardwareID) ON DELETE CASCADE ON
    UPDATE CASCADE;
174 GO
175
176 -- CHECK CONSTRAINTS
177 -- table: User
178 -- checks that the value of UserFirstName does not contain
```

```
178   numbers
179 ALTER TABLE [User]
180 ADD CONSTRAINT CK__User__UserFirstName
181 CHECK (UserFirstName NOT LIKE '%[0-9]%');
182 GO
183 -- checks that the values of UserPhoneNum are numbers
184 ALTER TABLE [User]
185 ADD CONSTRAINT CK__User__UserPhoneNum
186 CHECK (UserPhoneNum LIKE '%[0-9]%');
187 GO
188 -- checks that the UserLastName contains a value and does
    not contain numbers
189 ALTER TABLE [User]
190 ADD CONSTRAINT CK__User__UserLastName
191 CHECK (LEN(UserLastName) > 0 AND UserLastname NOT LIKE '%[
    0-9]%');
192 GO
193
194 -- table: Staff
195 -- checks that the value of StaffFirstName does not
    contain numbers
196 ALTER TABLE Staff
197 ADD CONSTRAINT CK__Staff__StaffFirstName
198 CHECK (StaffFirstName NOT LIKE '%[0-9]%');
199 GO
200 -- checks that StaffLastName contains a value and does not
     contain numbers
201 ALTER TABLE Staff
202 ADD CONSTRAINT CK__Staff__StaffLastName
203 CHECK (LEN(StaffLastName) > 0 AND StaffLastname NOT LIKE
    '%[0-9]%');
204 GO
205
206 -- table: TicketCategory
207 -- checks that CategoryName contains a value
208 ALTER TABLE TicketCategory
209 ADD CONSTRAINT CK__TicketCategory__CategoryName
210 CHECK (LEN(CategoryName) > 0 AND CategoryName NOT LIKE '%[
    0-9]%');
211 GO
212
213 -- table: Task
214 -- checks that TaskInformation contains a value
215 ALTER TABLE Task
216 ADD CONSTRAINT CK__Task__TaskInformation
217 CHECK (LEN(TaskInformation) > 0);
```

```
218 GO
219
220
221 -- UNIQUE CONSTRAINTS
222 -- table: User
223 ALTER TABLE [User]
224 ADD CONSTRAINT AK__User__UserPhoneNum
225 UNIQUE (UserPhoneNum);
226 GO
227
228 -- table: Hardware
229 ALTER TABLE Hardware
230 ADD CONSTRAINT AK__Hardware__SerialNumber
231 UNIQUE (SerialNumber);
232 GO
233
234
235 -- DEFAULT CONSTRAINTS
236 -- table: User
237 ALTER TABLE [User]
238 ADD CONSTRAINT DF__User__UserFirstName
239 DEFAULT ('UserFirstName') FOR UserFirstName;
240 GO
241
242 -- table: Staff
243 ALTER TABLE Staff
244 ADD CONSTRAINT DF__Staff__StaffFirstName
245 DEFAULT ('StaffFirstName') FOR StaffFirstName;
246 GO
247 -- adding the CommentDate
248 ALTER TABLE Comment
249 ADD CommentDate DATE NOT NULL DEFAULT (GETDATE());
250 GO
251 -- setting the CommentMade's default value
252 ALTER TABLE Comment
253 ADD CONSTRAINT DF__Comment__CommentMade
254 DEFAULT ('') FOR CommentMade;
255 GO
256 -- setting the DateSolved's default value
257 ALTER TABLE Task
258 ADD DateSolved DATE NOT NULL DEFAULT (GETDATE());
259 GO
260
261 -- CREATE INDEX
262 -- table: User
263 CREATE INDEX IX__User__UserLastName
```

```
264 ON [User] (UserLastName);
265 CREATE INDEX IX__User__UserFirstName__UserLastName
266 ON [User] (UserLastName DESC, UserFirstName ASC);
267
268 -- table: Staff
269 CREATE INDEX IX__Staff__StaffLastName
270 ON Staff (StaffLastName);
271 CREATE INDEX IX__Staff__FirstName__LastName
272 ON Staff (StaffLastName DESC, StaffFirstName ASC)
273 GO
274
275 -- table: Ticket
276 CREATE INDEX IX__Ticket__Issue
277 ON Ticket (Issue);
278
279 -- table: TicketCategory
280 CREATE INDEX IX__TicketCategory__CategoryDescription
281 ON TicketCategory (CategoryDescription);
282
283 -- table: Task
284 CREATE INDEX IX__Task__TaskInformation
285 ON Task (TaskInformation);
286
287 -- table: Hardware
288 CREATE INDEX IX__Hardware__ModelNumber
289 ON Hardware (SerialNumber);
290
291
```