

```
1 USE master;
2 GO
3
4
5 -- create the database
6 DROP DATABASE IF EXISTS SpeedingTickets;
7 CREATE DATABASE SpeedingTickets;
8 GO
9
10
11 USE SpeedingTickets;
12 GO
13
14
15 -- create the tables
16 DROP TABLE IF EXISTS "User";
17 CREATE TABLE "User"
18 (
19     UserID      INT IDENTITY NOT NULL,
20     FirstName   NVARCHAR(32) NOT NULL,
21     LastName    NVARCHAR(32) NOT NULL,
22     Photo       IMAGE        NULL,
23     PhoneNumber NCHAR(11)   NOT NULL,
24     EmailAddress NVARCHAR(32) NOT NULL,
25     BirthDate   DATE        NOT NULL,
26     Username    NVARCHAR(32) NOT NULL,
27     Password    NVARCHAR(32) NOT NULL,
28     PRIMARY KEY (UserID)
29 );
30
31 DROP TABLE IF EXISTS Staff;
32 CREATE TABLE Staff
33 (
34     StaffID     INT IDENTITY NOT NULL,
35     FirstName   NVARCHAR(32) NOT NULL,
36     LastName    NVARCHAR(32) NOT NULL,
37     Photo       IMAGE        NULL,
38     PhoneNumber NCHAR(11)   NOT NULL,
39     EmailAddress NVARCHAR(32) NOT NULL,
40     BirthDate   DATE        NOT NULL,
41     Username    NVARCHAR(32) NOT NULL,
42     Password    NVARCHAR(32) NOT NULL,
43     PRIMARY KEY (StaffID)
44 );
45
46 DROP TABLE IF EXISTS Hardware;
```

```

47 CREATE TABLE Hardware
48 (
49     HardwareID      INT IDENTITY NOT NULL,
50     Description    NVARCHAR(80) NOT NULL,
51     Category        NVARCHAR(32) NOT NULL,
52     Brand           NVARCHAR(32) NOT NULL,
53     AcquisitionDate DATE          NOT NULL,
54     WarrantyDate   DATE          NOT NULL,
55     SerialCode      NVARCHAR(32) NOT NULL,
56     PRIMARY KEY (HardwareID)
57 );
58
59 DROP TABLE IF EXISTS Category;
60 CREATE TABLE Category
61 (
62     CategoryID     INT IDENTITY NOT NULL,
63     Description    NVARCHAR(32) NOT NULL,
64     SLA            NVARCHAR(32) NOT NULL,
65     PRIMARY KEY (CategoryID)
66 );
67
68 DROP TABLE IF EXISTS Ticket;
69 CREATE TABLE Ticket
70 (
71     TicketID       INT IDENTITY NOT NULL,
72     Title          NVARCHAR(80) NOT NULL,
73     Description    NVARCHAR(255) NOT NULL,
74     StartDate      DATETIME2 NOT NULL,
75     SLA            DATETIME2 NOT NULL,
76     Status          NVARCHAR(32) NOT NULL,
77     EndDate         DATETIME2 NULL,
78     UserID          INT          NULL,
79     CategoryID     INT          NOT NULL,
80     HardwareID     INT          NULL,
81     PRIMARY KEY (TicketID)
82 );
83
84 DROP TABLE IF EXISTS Comment;
85 CREATE TABLE Comment
86 (
87     CommentID      INT IDENTITY NOT NULL,
88     Description    NVARCHAR(255) NOT NULL,
89     Date           DATETIME2 NOT NULL,
90     TicketID       INT          NOT NULL,
91     UserID          INT          NULL,
92     StaffID        INT          NULL,

```

```

93      PRIMARY KEY (CommentID)
94  );
95
96  DROP TABLE IF EXISTS Task;
97  CREATE TABLE Task
98  (
99      TaskID      INT IDENTITY NOT NULL,
100     Description NVARCHAR(255) NOT NULL,
101     StartDate   DATETIME2    NOT NULL,
102     EndDate    DATETIME2    NULL,
103     TicketID   INT          NOT NULL,
104     StaffID    INT          NULL,
105     PRIMARY KEY (TaskID)
106  );
107
108  DROP TABLE IF EXISTS TicketStaff;
109  CREATE TABLE TicketStaff
110  (
111     StaffID      INT          NOT NULL,
112     TicketID    INT          NOT NULL,
113     AssignedDate DATETIME2    NOT NULL
114  );
115
116
117  -- create relationships
118  ALTER TABLE Ticket
119      DROP CONSTRAINT IF EXISTS FK__User__UserID_T;
120  ALTER TABLE Ticket
121      ADD CONSTRAINT FK__User__UserID_T
122          FOREIGN KEY (UserID) REFERENCES "User" (UserID)
123          ON DELETE SET NULL ON UPDATE CASCADE;
124
125  ALTER TABLE Ticket
126      DROP CONSTRAINT IF EXISTS FK__Category__CategoryID_T;
127  ALTER TABLE Ticket
128      ADD CONSTRAINT FK__Category__CategoryID_T
129          FOREIGN KEY (CategoryID) REFERENCES Category (
130              CategoryID)
131          ON DELETE NO ACTION ON UPDATE CASCADE;
132
133  ALTER TABLE Ticket
134      DROP CONSTRAINT IF EXISTS FK__Hardware__HardwareID_T;
135  ALTER TABLE Ticket
136      ADD CONSTRAINT FK__Hardware__HardwareID_T
137          FOREIGN KEY (HardwareID) REFERENCES Hardware (
138              HardwareID)

```

```

137      ON DELETE SET NULL ON UPDATE CASCADE;
138
139
140 ALTER TABLE Comment
141     DROP CONSTRAINT IF EXISTS FK__Ticket__TicketID_C;
142 ALTER TABLE Comment
143     ADD CONSTRAINT FK__Ticket__TicketID_C
144         FOREIGN KEY (TicketID) REFERENCES Ticket (TicketID)
145         ON DELETE CASCADE ON UPDATE CASCADE;
146
147 ALTER TABLE Comment
148     DROP CONSTRAINT IF EXISTS FK__User__UserID_C;
149 ALTER TABLE Comment
150     ADD CONSTRAINT FK__User__UserID_C
151         FOREIGN KEY (UserID) REFERENCES "User" (UserID);
152         --ON DELETE SET NULL ON UPDATE CASCADE; It was
153             necessary to comment this line So that the error "multiple
154             cascade paths" did not occur but now it is impossible to
155             delete a user who has a ticket and comment
156
157 ALTER TABLE Comment
158     DROP CONSTRAINT IF EXISTS FK__Staff__StaffID_C;
159 ALTER TABLE Comment
160     ADD CONSTRAINT FK__Staff__StaffID_C
161         FOREIGN KEY (StaffID) REFERENCES Staff (StaffID)
162         ON DELETE SET NULL ON UPDATE CASCADE;
163
164
165 ALTER TABLE Task
166     DROP CONSTRAINT IF EXISTS FK__Ticket__TicketID_Ta;
167 ALTER TABLE Task
168     ADD CONSTRAINT FK__Ticket__TicketID_Ta
169         FOREIGN KEY (TicketID) REFERENCES Ticket (TicketID)
170         ON DELETE CASCADE ON UPDATE CASCADE;
171
172 ALTER TABLE Task
173     DROP CONSTRAINT IF EXISTS FK__Staff__StaffID_Ta;
174 ALTER TABLE Task
175     ADD CONSTRAINT FK__Staff__StaffID_Ta
176         FOREIGN KEY (StaffID) REFERENCES Staff (StaffID)
177         ON DELETE SET NULL ON UPDATE CASCADE;
178
179 -- create composite key in the TicketStaff table
180 ALTER TABLE TicketStaff
181     DROP CONSTRAINT IF EXISTS PK__TicketStaff;

```

```

180 ALTER TABLE TicketStaff
181     ADD CONSTRAINT PK__TicketStaff
182         PRIMARY KEY (StaffID, TicketID);
183
184 ALTER TABLE TicketStaff
185     DROP CONSTRAINT IF EXISTS FK__Staff_StaffID_TS;
186 ALTER TABLE TicketStaff
187     ADD CONSTRAINT FK__Staff_StaffID_TS
188         FOREIGN KEY (StaffID) REFERENCES Staff (StaffID)
189         ON DELETE CASCADE ON UPDATE CASCADE;
190
191 ALTER TABLE TicketStaff
192     DROP CONSTRAINT IF EXISTS FK__Ticket_TicketID_TS;
193 ALTER TABLE TicketStaff
194     ADD CONSTRAINT FK__Ticket_TicketID_TS
195         FOREIGN KEY (TicketID) REFERENCES Ticket (TicketID)
196         ON DELETE CASCADE ON UPDATE CASCADE;
197
198
199 -- create check constraint
200 ALTER TABLE "User"
201     DROP CONSTRAINT IF EXISTS CK__User__EmailAddress;
202 ALTER TABLE "User"
203     ADD CONSTRAINT CK__User__EmailAddress CHECK (
204         EmailAddress LIKE '%@%.%');
205
206 ALTER TABLE "User"
207     DROP CONSTRAINT IF EXISTS CK__User__Password;
208 ALTER TABLE "User"
209     ADD CONSTRAINT CK__User__Password CHECK (Password LIKE
210         '[A-Z][A-Z][A-Z][A-Z][A-Z][A-Z][0-9][0-9]');
211
212 ALTER TABLE Staff
213     DROP CONSTRAINT IF EXISTS CK__Staff__EmailAddress;
214 ALTER TABLE Staff
215     ADD CONSTRAINT CK__Staff__EmailAddress CHECK (
216         EmailAddress LIKE '%@%.%');
217
218 ALTER TABLE Staff
219     DROP CONSTRAINT IF EXISTS CK__Staff__Password;
220 ALTER TABLE Staff
221     ADD CONSTRAINT CK__Staff__Password CHECK (Password
222         LIKE '[A-Z][A-Z][A-Z][A-Z][A-Z][A-Z][0-9][0-9]');

```

```

222 -- create unique constraint
223 ALTER TABLE "User"
224     DROP CONSTRAINT IF EXISTS UQ__User__Username;
225 ALTER TABLE "User"
226     ADD CONSTRAINT UQ__User__Username UNIQUE (Username);
227
228 ALTER TABLE Staff
229     DROP CONSTRAINT IF EXISTS UQ__Staff__Username;
230 ALTER TABLE Staff
231     ADD CONSTRAINT UQ__Staff__Username UNIQUE (Username);
232
233
234 -- create default constraint
235 ALTER TABLE "User"
236     DROP CONSTRAINT IF EXISTS DF__User__Password;
237 ALTER TABLE "User"
238     ADD CONSTRAINT DF__User__Password DEFAULT ('ABCDEF12'
239 ) FOR Password;
240
241 ALTER TABLE Staff
242     DROP CONSTRAINT IF EXISTS DF__Staff__Password;
243 ALTER TABLE Staff
244     ADD CONSTRAINT DF__Staff__Password DEFAULT ('ABCDEF12'
245 ) FOR Password;
246
247 ALTER TABLE Hardware
248     DROP CONSTRAINT IF EXISTS
249 DF__Hardware__AcquisitionDate;
250 ALTER TABLE Hardware
251     ADD CONSTRAINT DF__Hardware__AcquisitionDate DEFAULT (
252 GetDate()) FOR AcquisitionDate;
253
254 ALTER TABLE Ticket
255     DROP CONSTRAINT IF EXISTS DF__Ticket__StartDate;
256 ALTER TABLE Ticket
257     ADD CONSTRAINT DF__Ticket__StartDate DEFAULT (GetDate(
258 )) FOR StartDate;
259
260 ALTER TABLE Ticket
261     DROP CONSTRAINT IF EXISTS DF__Ticket__Status;
262 ALTER TABLE Ticket
263     ADD CONSTRAINT DF__Ticket__Status DEFAULT ('New') FOR
Status;

```

```
262  
263  
264 ALTER TABLE Comment  
265     DROP CONSTRAINT IF EXISTS DF__Comment__Date;  
266 ALTER TABLE Comment  
267     ADD CONSTRAINT DF__Comment__Date DEFAULT (GetDate())  
268     FOR Date;  
269  
270 ALTER TABLE TicketStaff  
271     DROP CONSTRAINT IF EXISTS  
272         DF__TicketStaff__AssignedDate;  
273 ALTER TABLE TicketStaff  
274     ADD CONSTRAINT DF__TicketStaff__AssignedDate DEFAULT (  
275         GetDate()) FOR AssignedDate;  
276  
277 -- create indexes to non-key fields used for Searching  
278 records  
279 CREATE INDEX IX__User__FirstNameLastName  
280     ON "User" (FirstName, LastName DESC);  
281  
282  
283  
284 CREATE INDEX IX__Staff__FirstNameLastName  
285     ON Staff (FirstName, LastName DESC);  
286  
287 CREATE INDEX IX__Staff__Username  
288     ON Staff (Username);  
289  
290  
291 CREATE INDEX IX__Hardware__SerialCode  
292     ON Hardware (SerialCode);  
293  
294 GO  
295  
296  
297 -- Assignment 3 Starts here!!!  
298 -- INSERT Statements  
299 SET IDENTITY_INSERT "User" ON; --allows a record to  
300 contain a value in the identity field during an insert  
301 INSERT INTO "User"  
302     (UserID, FirstName, LastName, Photo, PhoneNumber,  
303      EmailAddress, BirthDate, Username, Password)
```

```

302 VALUES
303     (1 , 'Joao' , 'Machado' , NULL , '17828821111' , '
      joao.machado@speed.com' , '1982-12-31' , 'joaomachado'
      , 'ABCDEF12'),
304     (2 , 'Paulo' , 'Oliveira' , NULL , '17828512525' , '
      paulo.oliveira@speed.com' , '1988-05-15' , '
      paulooliveira' , 'ABCDEF12'),
305     (3 , 'Caio' , 'Vieira' , NULL , '17826665555' , '
      caio.vieira@speed.com' , '1992-01-11' , 'caiovieira'
      , 'ABCDEF12'),
306     (4 , 'Pedro' , 'Silva' , NULL , '17821239876' , '
      pedro.silva@speed.com' , '1995-03-09' , 'pedrosilva'
      , 'ABCDEF12'),
307     (5 , 'Gustavo' , 'Almeida' , NULL , '17829875463' , '
      gustavo.almeida@speed.com' , '1979-12-21' , '
      gustavoalmeida' , 'ABCDEF12'),
308     (6 , 'Carlos' , 'Porto' , NULL , '17829996666' , '
      carlos.porto@speed.com' , '1990-11-19' , 'carlosporto'
      , 'ABCDEF12'),
309     (7 , 'Maria' , 'Aparecida' , NULL , '17823215656' , '
      maria.aparecida@speed.com' , '1991-10-31' , '
      mariaaparecida' , 'ABCDEF12'),
310     (8 , 'Fernanda' , 'Goncalves' , NULL , '17827771515' , '
      fernanda.goncalves@speed.com' , '1987-07-06' , '
      fernandagoncalves' , 'ABCDEF12'),
311     (9 , 'Daniel' , 'Dalago' , NULL , '17826563232' , '
      daniel.dalago@speed.com' , '1992-10-31' , 'danieldalago'
      , 'ABCDEF12'),
312     (10 , 'Cristiane' , 'Pereira' , NULL , '17825552211' , '
      cristiane.pereira@speed.com' , '1996-04-12' , '
      cristianepereira' , 'ABCDEF12');
313 SET IDENTITY_INSERT "User" OFF;
314
315 SET IDENTITY_INSERT Staff ON;
316 INSERT INTO Staff
317     (StaffID, FirstName, LastName, Photo, PhoneNumber,
      EmailAddress, BirthDate, Username, Password)
318 VALUES
319     (1 , 'Gabriel' , 'Azevedo' , NULL , '17828821111' , '
      gabriel.azevedo@speed.com' , '1982-12-31' , '
      gabrielazevedo' , 'ABCDEF12'),
320     (2 , 'Juliana' , 'Ferreira' , NULL , '17828512525' , '
      juliana.ferreira@speed.com' , '1988-05-15' , '
      julianaferreira' , 'ABCDEF12'),
321     (3 , 'Alexandre' , 'Porto' , NULL , '17826665555' , '
      alexandre.porto@speed.com' , '1992-01-11' , '

```

```

321 alexandreporto',      'ABCDEF12'),  

322      (4 , 'Edson',      'Cechet',     NULL, '17821239876', '  

   edson.cechet@speed.com',      '1995-03-09', 'edsoncechet'  

   ,      'ABCDEF12'),  

323      (5 , 'Vanessa',    'Almeida',    NULL, '17829875463', '  

   vanessa.almeida@speed.com', '1979-12-21', '  

   vanessaalmeida',      'ABCDEF12'),  

324      (6 , 'Luciane',    'Ody',        NULL, '17829996666', '  

   luciane.ody@speed.com', '1990-11-19', 'lucianeody'  

   ,      'ABCDEF12'),  

325      (7 , 'Bruno',      'Fischer',    NULL, '17823215656', '  

   bruno.fischer@speed.com', '1991-10-31', 'brunofischer'  

   ,      'ABCDEF12'),  

326      (8 , 'Patricia',   'Goncalves',  NULL, '17827771515', '  

   patricia.goncalves@speed.com', '1987-07-06', '  

   patriciagoncalves',      'ABCDEF12'),  

327      (9 , 'Rodrigo',    'Dalago',     NULL, '17826563232', '  

   rodrigo.dalago@speed.com', '1992-10-31', '  

   rodrigodalago',      'ABCDEF12'),  

328      (10, 'Marcel',     'Pereira',    NULL, '17825552211', '  

   marcel.pereira@speed.com', '1996-04-12', '  

   marcelpereira',      'ABCDEF12'));  

329 SET IDENTITY_INSERT Staff OFF;  

330  

331 SET IDENTITY_INSERT Hardware ON;  

332 INSERT INTO Hardware  

333      (HardwareID, Description, Category, Brand,  

   AcquisitionDate, WarrantyDate, SerialCode)  

334 VALUES  

335      (1,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500001'),  

336      (2,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500002'),  

337      (3,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500003'),  

338      (4,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500004'),  

339      (5,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500005'),  

340      (6,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500006'),  

341      (7,  'Desktop Dell i5 16GB RAM 1TB', 'Desktop',  'Dell  

   ',      '2018-12-05', '2020-12-05', 'D016R1HDI500007'),  

342      (8,  'Mouse Microsoft 5000bpi',      'Mouse',    '  

   Microsoft', '2017-06-01', '2021-06-01', 'M0U53MS5000B001'  

   ),
```

```

343      (9, 'Mouse Microsoft 5000bpi', 'Mouse', 'Microsoft', '2017-06-01', '2021-06-01', 'M0U53MS5000B002')
),
344      (10, 'Keyboard Microsoft', 'Keyboard', 'Microsoft', '2019-05-20', '2022-05-20', 'K3YB0MS0FT00001'
);
345 SET IDENTITY_INSERT Hardware OFF;
346
347 SET IDENTITY_INSERT Category ON;
348 INSERT INTO Category
349     (CategoryID, Description, SLA)
350 VALUES
351     (1, 'Hardware Peripheral', '7 days'),
352     (2, 'Hardware Desktop', '2 days'),
353     (3, 'Microsoft Office', '3 days'),
354     (4, 'SAP', '1 day'),
355     (5, 'Oracle Database', '6 hours'),
356     (6, 'Access Problem', '6 hours'),
357     (7, 'Internet Problem', '1 day'),
358     (8, 'Software Installation', '2 days'),
359     (9, 'Hardware Monitor', '4 days'),
360     (10, 'Other Software', '3 days');
361 SET IDENTITY_INSERT Category OFF;
362
363 SET IDENTITY_INSERT Ticket ON;
364 INSERT INTO Ticket
365     (TicketID, Title, Description, StartDate, SLA, Status
, EndDate, UserID, CategoryID, HardwareID)
366 VALUES
367     (1, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-08 14:
15:10', '2020-03-10 14:15:10', 'New', NULL
, 1, 2, 2),
368     (2, 'SAP issue', 'SAP does not work.'
, '2020-03-08 14:15
:10', '2020-03-10 14:15:10', 'Open', NULL
, 4, 4, NULL),
369     (3, 'MS Word issue', 'Microsoft Word does
not work.', '2020-02-02 14:15:
10', '2020-02-10 14:15:10', 'Closed', '2020-02-09 14:15:10
', 4, 3, NULL),
370     (4, 'Mouse issue', 'Broken mouse.'
, '2020-03-03
14:15:10', '2020-03-10 14:15:10', 'Open', NULL
, NULL, 1, 9),
371     (5, 'Oracle DB issue', 'Oracle database is

```

```

371 unavailable.',                                '2020-02-08 14:15
   :10', '2020-02-10 14:15:10', 'Open ', NULL
   ,           NULL, 5, NULL),
372   (6, 'MS Excel issue',          'Microsoft Excel does
not work.',                               '2020-02-01 14:15:
10', '2020-02-10 14:15:10', 'New',    NULL
   ,           7, 3, NULL),
373   (7, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-02-02 14:
15:10', '2020-02-10 14:15:10', 'Open',   NULL
   ,           2, 7, 3),
374   (8, 'Internet issue',        'Computer without
internet access.',                      '2020-03-03 14:
15:10', '2020-03-10 14:15:10', 'Closed', '2020-03-09 14:15
:10', 9, 1, 4),
375   (9, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-04-11 14:
15:10', '2020-04-10 14:15:10', 'Open',   NULL
   ,           8, 1, 6),
376   (10, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-12 14:
15:10', '2020-03-10 14:15:10', 'New',    NULL
   ,           6, 6, 6),
377   (11, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-13 14:
15:10', '2020-03-10 14:15:10', 'New',    NULL
   ,           1, 2, 2),
378   (12, 'SAP issue',            'SAP does not work.'
   ,                           '2020-03-20 14:15
:10', '2020-03-10 14:15:10', 'Open',   NULL
   ,           4, 4, NULL),
379   (13, 'MS Word issue',       'Microsoft Word does
not work.',                             '2020-04-21 14:15:
10', '2020-04-10 14:15:10', 'Closed', '2020-04-09 14:15:10
', 4, 3, NULL),
380   (14, 'Mouse issue',         'Broken mouse.'
   ,                           '2020-03-22
14:15:10', '2020-03-10 14:15:10', 'Open',   NULL
   ,           NULL, 1, 9),
381   (15, 'Oracle DB issue',     'Oracle database is
unavailable.',                            '2020-03-23 14:15
:10', '2020-03-10 14:15:10', 'Open ', NULL
   ,           NULL, 5, NULL),
382   (16, 'MS Excel issue',     'Microsoft Excel does
not work.',                             '2020-03-31 14:15:
10', '2020-03-10 14:15:10', 'New',    NULL

```

```

382 , 7, 3, NULL),
383 (17, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-04-30 14:
15:10', '2020-04-10 14:15:10', 'Open', NULL
, 2, 7, 3),
384 (18, 'Internet issue', 'Computer without
internet access.', '2020-03-08 14:
15:10', '2020-03-10 14:15:10', 'Closed', '2020-03-09 14:15
:10', 9, 1, 4),
385 (19, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-08 14:
15:10', '2020-03-10 14:15:10', 'Open', NULL
, 8, 1, 6),
386 (20, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-08 14:
15:10', '2020-03-10 14:15:10', 'New', NULL
, 6, 6, 6),
387 (21, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-04-08 14:
15:10', '2020-04-10 14:15:10', 'New', NULL
, 1, 2, 2),
388 (22, 'SAP issue', 'SAP does not work.'
, '2020-03-08 14:15
:10', '2020-03-10 14:15:10', 'Open', NULL
, 4, 4, NULL),
389 (23, 'MS Word issue', 'Microsoft Word does
not work.', '2020-03-02 14:15
:10', '2020-03-10 14:15:10', 'Closed', '2020-03-09 14:15:10
', 4, 3, NULL),
390 (24, 'Mouse issue', 'Broken mouse.'
, '2020-03-03
14:15:10', '2020-03-10 14:15:10', 'Open', NULL
, NULL, 1, 9),
391 (25, 'Oracle DB issue', 'Oracle database is
unavailable.', '2020-03-08 14:15
:10', '2020-03-10 14:15:10', 'Open ', NULL
, NULL, 5, NULL),
392 (26, 'MS Excel issue', 'Microsoft Excel does
not work.', '2020-03-08 14:15
:10', '2020-03-10 14:15:10', 'New', NULL
, 7, 3, NULL),
393 (27, 'Computer does not work', 'Computer does not
Start when the power button is pressed.', '2020-03-08 14:
15:10', '2020-03-10 14:15:10', 'Open', NULL
, 3, 1, 3),
394 (28, 'Internet issue', 'Computer without

```

```

394    internet access.',          '2020-04-08 14:15:10', '2020-04-09 14:15:10', 9, 9, 4),
395    (29, 'Computer does not work', 'Computer does not Start when the power button is pressed.', '2020-04-08 14:15:10', '2020-04-10 14:15:10', 'Open', NULL
396    , 8, 1, 6),
397    (30, 'Computer does not work', 'Computer does not Start when the power button is pressed.', '2020-04-08 14:15:10', '2020-04-10 14:15:10', 'New', NULL
398    , 6, 10, 6);
399
400 SET IDENTITY_INSERT Ticket OFF;
401
402 INSERT INTO Comment
403     (CommentID, Description, Date, TicketID, UserID,
404      StaffID)
405 VALUES
406     (1, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 1, NULL, 2),
407     (2, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 2, NULL, 3),
408     (3, 'First attempt to contact by phone but user did not answer.', '2020-02-08 15:15:10', 2, NULL, 4),
409     (4, 'First attempt to contact by phone but initial comment.', '2020-03-08 15:15:10', 2, NULL, 1),
410     (5, 'First attempt to contact by phone but user did not answer.', '2020-02-08 15:15:10', 3, NULL, 2),
411     (6, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 3, NULL, 3),
412     (7, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 3, NULL, 4),
413     (8, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 4, NULL, 1),
414     (9, 'I am Sorry. Could you phone me now? I am at my desk now.', '2020-02-08 15:15:10', 4, 2, NULL),
415     (10, 'I am Sorry. Could you phone me now? initial comment.', '2020-03-08 15:15:10', 5, 4, NULL),
416     (11, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 5, NULL, 2),
417     (12, 'First attempt to contact by phone but user did not answer.', '2020-03-08 15:15:10', 6, NULL, 3),
418     (13, 'First attempt to contact by phone but user did not answer.', '2020-02-08 15:15:10', 7, NULL, 4),
419     (14, 'First attempt to contact by phone but initial

```

```

417 comment.', '2020-04-08 15:15:10', 8, NULL, 1),
418     (15, 'First attempt to contact by phone but user did
not answer.', '2020-03-08 15:15:10', 8, NULL, 2),
419     (16, 'First attempt to contact by phone but user did
not answer.', '2020-02-08 15:15:10', 9, NULL, 3),
420     (17, 'First attempt to contact by phone but user did
not answer.', '2020-03-08 15:15:10', 10, NULL, 4),
421     (18, 'First attempt to contact by phone but user did
not answer.', '2020-02-08 15:15:10', 10, NULL, 1),
422     (19, 'I am Sorry. Could you phone me now? I am at my
desk now.', '2020-03-08 15:15:10', 11, 2, NULL),
423     (20, 'I am Sorry. Could you phone me now? initial
comment.', '2020-03-08 15:15:10', 12, 4, NULL),
424     (21, 'First attempt to contact by phone but user did
not answer.', '2020-02-08 15:15:10', 13, NULL, 2),
425     (22, 'First attempt to contact by phone but user did
not answer.', '2020-03-08 15:15:10', 10, NULL, 3),
426     (23, 'First attempt to contact by phone but user did
not answer.', '2020-04-08 15:15:10', 10, NULL, 4),
427     (24, 'First attempt to contact by phone but initial
comment.', '2020-03-08 15:15:10', 11, NULL, 1),
428     (25, 'First attempt to contact by phone but user did
not answer.', '2020-04-08 15:15:10', 12, NULL, 2),
429     (26, 'First attempt to contact by phone but user did
not answer.', '2020-03-08 15:15:10', 13, NULL, 3),
430     (27, 'First attempt to contact by phone but user did
not answer.', '2020-04-08 15:15:10', 23, NULL, 4),
431     (28, 'First attempt to contact by phone but user did
not answer.', '2020-04-08 15:15:10', 24, NULL, 1),
432     (29, 'I am Sorry. Could you phone me now? I am at my
desk now.', '2020-04-08 15:15:10', 25, 2, NULL),
433     (30, 'I am Sorry. Could you phone me now? initial
comment.', '2020-04-08 15:15:10', 20, 4, NULL);
434 SET IDENTITY_INSERT Comment OFF;
435
436 SET IDENTITY_INSERT Task ON;
437 INSERT INTO Task
438     (TaskID, Description, StartDate, EndDate, TicketID,
StaffID)
439 VALUES
440     (1, 'Install Software', '2020-04-08 15:15:
10', '2020-04-09 15:15:10', 1, 2 ),
441     (2, 'Install a new power Supply', '2020-04-08 15:15:
10', null, 2, 2 ),
442     (3, 'Install Software', '2020-04-08 15:15:
10', null, 3, 3 ),

```

```

443      (4, 'Install a new power Supply', '2020-04-08 15:15:
444          10', null,           4, 4 ),
444      (5, 'Install Software',           '2020-04-08 15:15:
444          10', null,           5, 5 ),
445      (6, 'Install a new power Supply', '2020-04-08 15:15:
445          10', null,           6, 1 ),
446      (7, 'Install Software',           '2020-04-08 15:15:
446          10', '2020-04-09 15:15:10', 10, null),
447      (8, 'Install a new power Supply', '2020-04-08 15:15:
447          10', null,           8, 3 ),
448      (9, 'Install Software',           '2020-04-08 15:15:
448          10', null,           10, 2 ),
449      (10, 'Install a new power Supply', '2020-04-08 15:15:
449          10', '2020-04-09 15:15:10', 10, null),
450      (11, 'Install Software',           '2020-04-08 15:15:
450          10', '2020-04-09 15:15:10', 1, 10 ),
451      (12, 'Install a new power Supply', '2020-04-08 15:15:
451          10', null,           2, 2 ),
452      (13, 'Install Software',           '2020-04-08 15:15:
452          10', null,           3, 3 ),
453      (14, 'Install a new power Supply', '2020-04-08 15:15:
453          10', null,           4, 4 ),
454      (15, 'Install Software',           '2020-04-08 15:15:
454          10', null,           5, 5 ),
455      (16, 'Install a new power Supply', '2020-04-08 15:15:
455          10', null,           6, 1 ),
456      (17, 'Install Software',           '2020-04-08 15:15:
456          10', '2020-04-09 15:15:10', 10, null),
457      (18, 'Install a new power Supply', '2020-04-08 15:15:
457          10', null,           8, 3 ),
458      (19, 'Install Software',           '2020-04-08 15:15:
458          10', null,           10, 2 ),
459      (20, 'Install a new power Supply', '2020-04-08 15:15:
459          10', '2020-04-09 15:15:10', 10, null);
460 SET IDENTITY_INSERT Task OFF;
461
462 INSERT INTO TicketStaff
463     (StaffID, TicketID, AssignedDate)
464 VALUES
465     (1, 1, '2020-03-08 15:15:10'),
466     (2, 2, '2020-03-09 15:15:10'),
467     (3, 3, '2020-03-09 16:15:10'),
468     (4, 4, '2020-03-08 19:15:10'),
469     (5, 5, '2020-03-09 10:15:10'),
470     (6, 6, '2020-03-08 12:15:10'),
471     (7, 7, '2020-03-10 15:15:10'),

```

```
472      (1, 8, '2020-03-09 09:15:10'),  
473      (2, 1, '2020-03-09 18:15:10'),  
474      (3, 1, '2020-03-08 11:15:10');  
475  
476 GO  
477  
478  
479 -- DELETE Statements  
480 -- A. Delete an user that has created a ticket  
481 DELETE FROM "User"  
482 WHERE UserID = 1;  
483 GO  
484  
485 -- B. Delete a Support Staff member that is currently  
        assigned to a ticket  
486 DELETE FROM Staff  
487 WHERE StaffID = 1;  
488 GO  
489  
490 -- C. Delete a ticket that has a task and a comment  
491 DELETE FROM Ticket  
492 WHERE TicketID = 1;  
493 GO  
494  
495 -- D. Delete a hardware that has more than one ticket  
496 DELETE FROM Hardware  
497 WHERE HardwareID = 6;  
498 GO  
499  
500  
501 -- UPDATE Statements  
502 -- A. Assign a Staff member to a ticket  
503 UPDATE TicketStaff  
504 SET StaffID = 2  
505 WHERE StaffID = 7 AND TicketID = 7;  
506 GO  
507  
508 -- B. Change the Status of a ticket to completed (or the  
        like)  
509 UPDATE Ticket  
510 SET Status = 'Closed', EndDate = GetDate()  
511 WHERE TicketID = 9;  
512 GO  
513  
514 -- C. Change the user id (primary key), for a user that  
        has created tickets, to a new value
```

```

515 ALTER TABLE "User" DROP CONSTRAINT IF EXISTS
      UQ__User__Username;
516
517 SET IDENTITY_INSERT "User" ON;
518 INSERT INTO "User"
      (UserID, FirstName, LastName, Photo, PhoneNumber,
       EmailAddress, BirthDate, Username, Password)
519 SELECT 11, FirstName, LastName, Photo, PhoneNumber,
      EmailAddress, BirthDate, Username, Password
520 FROM "User"
521 WHERE UserID = 9;
522 SET IDENTITY_INSERT "User" OFF;
523
524
525 UPDATE Ticket
526 SET UserID = 11
527 WHERE UserID = 9;
528
529 UPDATE Comment
530 SET UserID = 11
531 WHERE UserID = 9;
532
533 DELETE FROM "User"
534 WHERE UserID = 9;
535
536 ALTER TABLE "User" ADD CONSTRAINT UQ__User__Username
      UNIQUE (Username);
537 GO
538
539
540 -- SELECT Statements
541 -- A. Retrieve a list of all open tickets
542 SELECT * FROM Ticket
543 WHERE EndDate IS NULL -- WHERE Status LIKE 'Open'
544 ORDER BY TicketID;
545 GO
546
547 -- B. Retrieve a list of tickets reported after Mar. 1,
      2020 at 4:30PM and before Mar. 4, 2020 at 8:30AM
548 SELECT * FROM Ticket
549 WHERE StartDate BETWEEN '2020-03-01 16:30:00' AND '2020-
      03-04 08:30:00'
550 ORDER BY StartDate;
551 GO
552
553 -- C. Retrieve a list of all tickets for three Specific
      categories

```

```

554 SELECT C.Description, T.*
555 FROM Category C INNER JOIN Ticket T ON C.CategoryID = T.
    CategoryID
556 WHERE C.Description = 'Hardware Peripheral' OR C.
    Description = 'Hardware Desktop' OR C.Description = '
        Microsoft Office'
557 ORDER BY C.Description;
558 GO
559
560 -- D. Retrieve a list of tickets assigned to a particular
     Staff member and ordered by assigned date
561 SELECT TS.AssignedDate, T.TicketID
562 FROM TicketStaff TS
563     LEFT OUTER JOIN Staff S
564         ON TS.StaffID = S.StaffID
565     RIGHT OUTER JOIN Ticket T
566         ON TS.TicketID = T.TicketID
567 WHERE S.FirstName = 'Juliana' AND S.LastName = 'Ferreira'
568 ORDER BY TS.AssignedDate;
569 GO
570
571 -- E. Retrieve a list of users' names and the number of
     tickets created by each user
572 -- Display zero (0) for those users that have not created
     any tickets and order the list by the user's last names
     .
573 SELECT U.FirstName, U.LastName, COUNT(T.TicketID) AS
    NumberOfTickets
574 FROM "User" U LEFT OUTER JOIN Ticket T ON U.UserID = T.
    UserID
575 GROUP BY U.FirstName, U.LastName
576 ORDER BY U.LastName;
577 GO
578
579 -- F. Retrieve a list of ticket titles that have comments
     with the phrase "initial comment" Somewhere in the text
580 SELECT T.Title, C.CommentID, C.Description
581 FROM Ticket T INNER JOIN Comment C ON T.TicketID = C.
    TicketID
582 WHERE C.Description LIKE '%initial comment%';
583 GO
584
585 -- G. Retrieve a list of the categories for tickets than
     have more than two (2) active tasks that have not been
     assigned to any Staff member
586 SELECT CT.Description, T.TicketID

```

```

587 FROM Ticket T
588     LEFT JOIN Category CT
589         ON T.CategoryID = CT.CategoryID
590     RIGHT JOIN Task TK
591         ON T.TicketID = TK.TicketID
592 WHERE TK.StaffID IS NULL
593 GROUP BY T.TicketID, CT.Description
594 HAVING COUNT(TK.TicketID) > 2;
595 GO
596
597
598 -- Assignment 4 Starts here!!!
599 -- 1. Display a list of all tickets submitted within a
     given month of the current year.
600 -- The month will be supplied to the routine as a word (e.
     g. April)
601
602 DROP PROCEDURE IF EXISTS usp_GetTicketsByMonth;
603 GO
604 CREATE PROCEDURE usp_GetTicketsByMonth
605     @month NVARCHAR(10)
606 AS
607 BEGIN
608     SELECT * From Ticket
609         WHERE DATENAME(MONTH,StartDate) LIKE @month AND YEAR(
             StartDate) = YEAR(GETDATE())
610         ORDER BY TicketID;
611 END;
612 GO
613
614 EXEC usp_GetTicketsByMonth N'February';
615 GO
616
617 EXEC usp_GetTicketsByMonth N'march';
618 GO
619
620 EXEC usp_GetTicketsByMonth N'APRIL';
621 GO
622
623 EXEC usp_GetTicketsByMonth N'may'; -- There is not any
     ticket in May and because of this there is not any
     selected ticket
624 GO
625
626
627 -- 2. Display a list of the top 10 tickets that have had

```

```

627 the most activity in the form
628 -- of comments. A start date and end date will be supplied
to the routine in the
629 -- form 'yyyy-mm-dd'.
630
631 DROP PROCEDURE IF EXISTS usp_GetTicketsWithMoreComments;
632 GO
633 CREATE PROCEDURE usp_GetTicketsWithMoreComments
634     @startDate NVARCHAR(10), @endDate NVARCHAR(10)
635 AS
636 BEGIN
637     SELECT TOP 10 T.TicketID, COUNT(C.CommentID) AS
        NumberofComments FROM Ticket T
638     INNER JOIN Comment C ON T.TicketID = C.TicketID
639     WHERE C.Date >= CONVERT(DATETIME, @startDate) AND C.
        Date <= CONVERT(DATETIME, @endDate)
640     GROUP BY T.TicketID
641     ORDER BY NumberofComments DESC;
642 END;
643 GO
644
645 EXEC usp_GetTicketsWithMoreComments N'2020-02-01', N'2020-
04-30';
646 GO
647
648 EXEC usp_GetTicketsWithMoreComments N'2020-04-01', N'2020-
04-30';
649 GO
650
651 EXEC usp_GetTicketsWithMoreComments N'2020-02-01', N'2020-
03-31';
652 GO
653
654 EXEC usp_GetTicketsWithMoreComments N'2020-03-10', N'2020-
03-31'; -- There is not any comment in that range and
because of this there is not any selected ticket
655 GO
656
657
658 -- 3. Display a list of tickets for a particular category
, ordered by descending date,
659 -- with the corresponding date displayed in the format (
Month dddd, yyyy) e.g.
660 -- November 21st, 2009. A category name will be supplied
to the routine.
661

```

```

662 DROP PROCEDURE IF EXISTS usp_GetTicketsByCategory;
663 GO
664 CREATE PROCEDURE usp_GetTicketsByCategory
665     @categoryName NVARCHAR(32)
666 AS
667 BEGIN
668     SELECT CT.Description, T.TicketID, DATENAME(MONTH,T.
       StartDate) +
669         CASE
670             WHEN DAY(T.StartDate) % 100 IN (11, 12, 13)
671             THEN CONCAT(' ',DAY(T.StartDate), 'th')
672             WHEN DAY(T.StartDate) % 10 = 1 THEN CONCAT('
   ,DAY(T.StartDate), 'st')
673             WHEN DAY(T.StartDate) % 10 = 2 THEN CONCAT('
   ,DAY(T.StartDate), 'nd')
674             WHEN DAY(T.StartDate) % 10 = 3 THEN CONCAT('
   ,DAY(T.StartDate), 'rd')
675             ELSE CONCAT(' ',DAY(T.StartDate), 'th') END
676         + FORMAT(T.StartDate, ', yyyy')
677     AS TicketDate FROM Category CT
678     INNER JOIN Ticket T ON CT.CategoryID = T.CategoryID
679     WHERE CT.Description LIKE @categoryName
680     ORDER BY T.StartDate DESC;
681 END;
682 GO
683
684 EXEC usp_GetTicketsByCategory N'Hardware Peripheral';
685 GO
686
687 EXEC usp_GetTicketsByCategory N'Hardware Desktop';
688 GO
689
690 EXEC usp_GetTicketsByCategory N'Microsoft Office';
691 GO
692
693 EXEC usp_GetTicketsByCategory N'SAP';
694 GO
695
696 EXEC usp_GetTicketsByCategory N'Oracle Database';
697 GO
698
699 EXEC usp_GetTicketsByCategory N'Access Problem';
700 GO
701
702 EXEC usp_GetTicketsByCategory N'Internet Problem';
703 GO

```

```
703
704 EXEC usp_GetTicketsByCategory N'Software Installation';
    -- No ticket in this Category because of this nothing is
    selected
705 GO
706
707 EXEC usp_GetTicketsByCategory N'Hardware Monitor';
708 GO
709
710 EXEC usp_GetTicketsByCategory N'Other Software';
711 GO
712
713 EXEC usp_GetTicketsByCategory N'Other Soft'; -- Category
    name incorrect because of this nothing is selected
714 GO
715
716
717 -- 4. Return the total number of active tasks for a given
    support staff member. An
718 -- employee number will be supplied to the routine. The
    routine should also
719 -- return zero (0), if there are no active tasks for that
    person, and negative one
720 -- (-1), if the support staff member could not be found.
721
722 DROP FUNCTION IF EXISTS
    ufn_GetNumberOfActiveTasksByStaffID;
723 GO
724 CREATE FUNCTION ufn_GetNumberOfActiveTasksByStaffID(@
    staffID INT)
725 RETURNS SMALLINT
726 AS
727 BEGIN
728     DECLARE @tasks SMALLINT;
729
730     IF EXISTS (SELECT * FROM Staff S
731                 WHERE S.StaffID = @staffID)
732         BEGIN
733             SET @tasks = (SELECT COUNT(TaskID) AS
    NumberOfActiveTasks FROM Task
734                         WHERE StaffID = @staffID AND
    EndDate IS NULL);
735         END
736
737     ELSE
738
```

```

739      BEGIN
740          SET @tasks = -1;
741      END
742      RETURN @tasks;
743 END;
744 GO
745
746 DECLARE @activeTasks SMALLINT;
747 DECLARE @staffID INT = 0;
748 DECLARE @name VARCHAR(32);
749 SELECT TOP 1 @staffID = StaffID, @name = CONCAT(FirstName
    , ' ', LastName)
750 FROM Staff
751 WHERE StaffID > @staffID
752 ORDER BY StaffID;
753 WHILE @@ROWCOUNT > 0
754 BEGIN
755     EXEC @activeTasks =
        ufn_GetNumberOfActiveTasksByStaffID @staffID;
756     PRINT 'StaffID ' + LTRIM(STR(@staffID)) + ' ' + @name
        + ' has ' + LTRIM(STR(@activeTasks)) + ' active tasks';
757     SELECT TOP 1 @staffID = StaffID, @name = CONCAT(
        FirstName, ' ', LastName)
758     FROM Staff
759     WHERE StaffID > @staffID
760     ORDER BY StaffID;
761 END
762 SET @staffID = 11;
763 EXEC @activeTasks = ufn_GetNumberOfActiveTasksByStaffID @
    staffID;
764 PRINT 'StaffID ' + LTRIM(STR(@staffID)) + ' has ' + LTRIM(
    STR(@activeTasks)) + ' active tasks';
765 GO
766
767
768 -- Another example using name
769 DROP FUNCTION IF EXISTS
    ufn_GetNumberOfActiveTasksByStaffName;
770 GO
771 CREATE FUNCTION ufn_GetNumberOfActiveTasksByStaffName(@
    firstName NVARCHAR(32), @lastName NVARCHAR(32))
772 RETURNS SMALLINT
773 AS
774 BEGIN
775     DECLARE @tasks SMALLINT;
776

```

```

777      IF EXISTS (SELECT * FROM Staff S
778          WHERE S.FirstName LIKE @firstName AND S.
779              LastName LIKE @lastName)
780          BEGIN
781              SET @tasks = (SELECT COUNT(TaskID) FROM Task T
782                  INNER JOIN Staff S ON T.StaffID =
783                      S.StaffID
784                  WHERE S.FirstName LIKE @firstName
785                  AND S.LastName LIKE @lastName AND T.EndDate IS NULL);
786          END
787      ELSE
788          BEGIN
789              SET @tasks = -1;
790          END
791      RETURN @tasks;
792  END;
793 GO
794 DECLARE @firstName NVARCHAR(32);
795 DECLARE @lastName NVARCHAR(32);
796 SET @firstName = 'Alexandre';
797 SET @lastName = 'Porto';
798 DECLARE @activeTasks SMALLINT;
799 EXEC @activeTasks = ufn_GetNumberOfActiveTasksByStaffName
    @firstName, @lastName;
800 PRINT @firstName + ' ' + @lastName + ' has ' + LTRIM(STR(@
    activeTasks)) + ' active tasks';
801 SET @firstName = 'Julian';
802 SET @lastName = 'Porto';
803 EXEC @activeTasks = ufn_GetNumberOfActiveTasksByStaffName
    @firstName, @lastName;
804 PRINT @firstName + ' ' + @lastName + ' has ' + LTRIM(STR(@
    activeTasks)) + ' active tasks';
805 GO
806
807
808 -- 5. Display a “page” of ticket information by passing,
     to the routine, a page
809 -- number and the number of tickets per page. For example
     , passing “1,10” will
810 -- return the first ten tickets (ordered by ticket id),
     but passing “2,10” will
811 -- return next ten tickets (i.e. page 2).
812

```

```
813 DROP PROCEDURE IF EXISTS usp_GetTicketsByPage;
814 GO
815 CREATE PROCEDURE usp_GetTicketsByPage
816     @PageNumber SMALLINT, @TicketsPerPage SMALLINT
817 AS
818 BEGIN
819     SELECT * FROM Ticket
820     ORDER BY TicketID
821     OFFSET ((@PageNumber - 1) * @TicketsPerPage) ROWS
822     FETCH NEXT @TicketsPerPage ROWS ONLY;
823 END;
824 GO
825
826 EXEC usp_GetTicketsByPage 1,10;
827 GO
828
829 EXEC usp_GetTicketsByPage 2,10;
830 GO
831
832 EXEC usp_GetTicketsByPage 3,5;
833 GO
834
835 EXEC usp_GetTicketsByPage 4,7;
836 GO
837
838 EXEC usp_GetTicketsByPage 10,2;
839 GO
840
841 EXEC usp_GetTicketsByPage 4,8;
842 GO
```