

```

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     -- checking constraint for phone number
25     CHECK (UserPhoneNum LIKE '[0-9][0-9][0-9][0-9][0-9]
26     ][0-9][0-9][0-9][0-9][0-9]'),
27     PRIMARY KEY (UserID)
28 );
29 GO
30
31 -- default constraint for User's First Name
32 ALTER TABLE [User]
33     DROP CONSTRAINT IF EXISTS DF__User__UserFirstName;
34 ALTER TABLE [User]
35     ADD CONSTRAINT DF__User__UserFirstName DEFAULT (') FOR
36     UserFirstName;
37
38 -- table: Staff
39 DROP TABLE IF EXISTS Staff;
40 CREATE TABLE Staff (
41     StaffID INT IDENTITY NOT NULL,
42     StaffFirstName VARCHAR(32) NOT NULL,
43     StaffLastName VARCHAR(32) NOT NULL,
44     StaffPhoneNum VARCHAR(32) NOT NULL,
45     StaffAddress VARCHAR(255) NOT NULL,
46     PRIMARY KEY (StaffID)

```

```
45 );
46 GO
47
48 -- table: Ticket
49 DROP TABLE IF EXISTS Ticket;
50 CREATE TABLE Ticket (
51     TicketID INT IDENTITY NOT NULL,
52     Issue VARCHAR(1024) NOT NULL,
53     DateIssued DATETIME NOT NULL,
54     DateResolved DATETIME NULL,
55     UserID INT NULL,
56     StaffID INT NULL,
57     StaffAssignedID INT NULL,
58     TicketCategoryID INT NOT NULL,
59     PRIMARY KEY (TicketID)
60 );
61 GO
62
63 -- table: TicketCategory
64 DROP TABLE IF EXISTS TicketCategory;
65 CREATE TABLE TicketCategory (
66     TicketCategoryID INT IDENTITY NOT NULL,
67     CategoryName VARCHAR(32) NOT NULL,
68     CategoryDescription VARCHAR(255) NOT NULL,
69     PRIMARY KEY (TicketCategoryID)
70 );
71 GO
72
73 -- table: Comment
74 DROP TABLE IF EXISTS Comment;
75 CREATE TABLE Comment (
76     CommentID INT IDENTITY NOT NULL,
77     CommentDate DATETIME NOT NULL,
78     CommentMade VARCHAR(1024) NOT NULL,
79     TicketID INT NOT NULL,
80     UserID INT NULL,
81     StaffID INT NULL,
82     PRIMARY KEY (CommentID)
83 );
84 GO
85
86 -- table: Task
87 DROP TABLE IF EXISTS Task;
88 CREATE TABLE Task (
89     TaskID INT IDENTITY NOT NULL,
90     TaskName VARCHAR(32) NOT NULL,
```

```
91     TaskInformation VARCHAR(255) NOT NULL,
92     DateTaskFinished DATETIME NULL,
93     TicketID INT NOT NULL,
94     StaffID INT NULL,
95     PRIMARY KEY (TaskID)
96 );
97 GO
98
99 -- table: Hardware
100 DROP TABLE IF EXISTS Hardware;
101 CREATE TABLE Hardware (
102     HardwareID INT IDENTITY NOT NULL,
103     HardwareName VARCHAR(32) NOT NULL,
104     HardwareDescription VARCHAR(80) NOT NULL,
105     PurchaseDate DATETIME NOT NULL,
106     SerialNumber VARCHAR(32) NOT NULL,
107     PRIMARY KEY (HardwareID)
108 );
109 GO
110
111 -- table: HardwareTicket
112 DROP TABLE IF EXISTS HardwareTicket;
113 CREATE TABLE HardwareTicket (
114     TicketID INT NOT NULL,
115     HardwareID INT NOT NULL,
116     PRIMARY KEY (TicketID, HardwareID)
117 );
118 GO
119
120 -- ALTER TABLES
121 -- foreign key constraints
122 -- table: Ticket
123 ALTER TABLE Ticket
124 ADD CONSTRAINT FK__Ticket__UserID
125 FOREIGN KEY (UserID)
126 REFERENCES [User] (UserID) ON DELETE SET NULL ON UPDATE
    SET NULL;
127 GO
128
129 ALTER TABLE Ticket
130 ADD CONSTRAINT FK__Ticket__StaffID
131 FOREIGN KEY (StaffID)
132 REFERENCES Staff (StaffID) ON DELETE SET NULL ON UPDATE
    SET NULL;
133 GO
134
```

```
135 ALTER TABLE Ticket
136 ADD CONSTRAINT FK__Ticket__StaffAssignedID
137 FOREIGN KEY (StaffAssignedID)
138 REFERENCES Staff (StaffID) ON DELETE NO ACTION ON UPDATE
    NO ACTION;
139 GO
140
141 ALTER TABLE Ticket
142 WITH CHECK ADD CONSTRAINT FK__Ticket__TicketCategoryID
143 FOREIGN KEY (TicketCategoryID)
144 REFERENCES TicketCategory (TicketCategoryID) ON DELETE
    CASCADE ON UPDATE CASCADE;
145 GO
146
147 -- table: Comment
148 ALTER TABLE Comment
149     DROP CONSTRAINT IF EXISTS FK__Comment__UserID;
150 ALTER TABLE Comment
151     ADD CONSTRAINT FK__Comment__User__UserID
152     FOREIGN KEY (UserID)
153     REFERENCES [User] (UserID);
154 GO
155
156 ALTER TABLE Comment
157     DROP CONSTRAINT IF EXISTS FK__Comment__TicketID;
158 ALTER TABLE Comment
159     ADD CONSTRAINT FK__Comment__TicketID
160     FOREIGN KEY (TicketID)
161     REFERENCES Ticket (TicketID);
162 GO
163
164 ALTER TABLE Comment
165     DROP CONSTRAINT IF EXISTS FK__Comment__StaffID;
166 ALTER TABLE Comment
167     WITH CHECK ADD CONSTRAINT FK__Comment__StaffID
168     FOREIGN KEY (StaffID)
169     REFERENCES [Staff] (StaffID);
170 GO
171
172 -- table: Task
173 ALTER TABLE Task
174     DROP CONSTRAINT IF EXISTS FK__Task__TicketID;
175 ALTER TABLE Task
176     ADD CONSTRAINT FK__Task__TicketID
177     FOREIGN KEY (TicketID)
178     REFERENCES Ticket (TicketID) ON DELETE CASCADE ON
```

```
178 UPDATE CASCADE;
179 GO
180
181 ALTER TABLE Task
182     DROP CONSTRAINT IF EXISTS FK__Task__StaffID;
183 ALTER TABLE Task
184     ADD CONSTRAINT FK__Task__StaffID
185     FOREIGN KEY (StaffID)
186     REFERENCES Staff (StaffID);
187 GO
188
189 -- table: HardwareTicket
190 ALTER TABLE HardwareTicket
191     DROP CONSTRAINT IF EXISTS
192     FK__HardwareTicket__HardwareID;
192 ALTER TABLE HardwareTicket
193     ADD CONSTRAINT FK__HardwareTicket__HardwareID
194     FOREIGN KEY (TicketID)
195     REFERENCES Ticket (TicketID);
196 GO
197
198 ALTER TABLE HardwareTicket
199     DROP CONSTRAINT IF EXISTS FK__Item__ItemID;
200 ALTER TABLE HardwareTicket
201 WITH CHECK ADD CONSTRAINT FK__Item__ItemID
202 FOREIGN KEY (HardwareID)
203 REFERENCES Hardware (HardwareID) ON DELETE CASCADE ON
    UPDATE CASCADE;
204 GO
205
206 -- INDEX STATEMENTS
207 CREATE INDEX IX__User__UserFirstNameLastName ON [User] (
    UserFirstName, UserLastName);
208 CREATE INDEX IX__Staff__StaffFirstNameLastName ON Staff (
    StaffFirstName, StaffLastName);
209 GO
210
211 CREATE INDEX IX_TicketCategory_CategoryDescription
212 ON TicketCategory (CategoryDescription);
213 GO
214
215 -- INSERT STATEMENTS
216 INSERT INTO [User] (UserFirstName, UserLastName,
    UserAddress, UserPhoneNum)
217 VALUES ('Chandler', 'Bing', '25 Place Avenue', '9024021111
    '),
```

```

218      ('Monica', 'Gellar-Bing', '77 Heavenly Way', '
9021777171'),
219      ('Ross', 'Gellar', '123 Dinosaur Parke', '
9025558888'),
220      ('Rachel', 'Green', '02 Fashion Avenue', '
9024447777'),
221      ('Joseph', 'Tribianni', '12 Howyoudoin court', '
9028557354'),
222      ('Phoebe', 'Buffay', '74 Smelly Cat Lane', '
9021458877');
223
224 INSERT INTO [Staff] (StaffFirstName, StaffLastName,
StaffPhoneNum, StaffAddress)
225 VALUES ('Matthew', 'Perry', '9025587789', '123 Address
Road'),
226      ('Courtney', 'Cox', '9021478526', '456 Location
Drive'),
227      ('David', 'Schwimmer', '9023114587', '789
Destination Highway'),
228      ('Jennifer', 'Anniston', '9021257448', '321 Area
Village'),
229      ('Matt', 'LeBlanc', '9021236547', '654 Pizza Road'
),
230      ('Lisa', 'Kudrow', '9024715544', '987 Music Way');
231
232 -- USING IDENTITY_INSERT TO OVERRIDE THE AUTOMATIC
IDENTITY GENERATION
233 SET IDENTITY_INSERT TicketCategory ON;
234 INSERT INTO TicketCategory (TicketCategoryID, CategoryName
, CategoryDescription)
235 VALUES (701, 'Hardware', 'There is an issue with hardware
or physical components'),
236      (702, 'Software', 'There is an issue with software
components'),
237      (703, 'Account', 'There is an issue with the user
account'),
238      (704, 'Internet', 'There is an issue with the
network connection'),
239      (705, 'General Questions', 'User needs help with
multiple general questions');
240 SET IDENTITY_INSERT TicketCategory OFF;
241
242 INSERT INTO Hardware (HardwareName, HardwareDescription,
PurchaseDate, SerialNumber)
243 VALUES ('MSI Optix', 'PC Monitor, Resolution: 1920x1080,
Refresh Rate: 144Hz, Aspect Ratio: 16:9', '2020-04-11 02:

```

```

243 30:00', 'MAH241C'),
244      ('Asus ROG Swift', 'PC Monitor, Resolution:
      3840x2160, Refresh Rate 144Hz, Aspect Ratio: 16:9', '2020-
      04-02 07:00:00', 'PG65UQ'),
245      ('Asus TUF Gaming', 'PC Monitor, Resolution:
      3840x2160, Refresh rate: 144Hz, Aspect Ratio: 16:9', '2020
      -01-20 02:00:00', 'VG279QM'),
246      ('Asus ROG Strix', 'PC Monitor, Resolution:
      1920x1080 Refresh Rate: 75Hz, Aspect Ratio: 16:9', '2020-
      02-14 02:00:00', 'XG279Q'),
247      ('Logitech Hero', 'Mouse, Connection Type: Wired,
      Maximum DPI: 16000, Color: Black', '2020-02-09 09:00:00',
      'G502'),
248      ('Logitech Lightspeed', 'Mouse, Connection Type:
      Wired, Maximum DPI: 16000, Color: Black', '2020-03-20 08:
      00:00', 'G702'),
249      ('Corsair Harpoon', 'Mouse, Connection Type: Wired
      , Maximum DPI: 16000, Color: Black', '2020-02-02 04:00:00'
      , 'AX213'),
250      ('Razer Naga Trinity', 'Mouse, Connection Type:
      Wireless, Maximum DPI: 14000, Color: White', '2020-02-21
      01:00:00', 'RZ123'),
251      ('SteelSeries Rival', 'Mouse, Connection Type:
      Wireless, Maximum DPI: 6000, Color: Pink', '2020-04-02 04:
      00:00', 'SS321'),
252      ('Razer Basilisk', 'Mouse, Connection Type:
      Wireless, Maximum DPI: 12000, Color: Blue', '2020-01-12 02
      :04:00', 'RZB744');
253
254 INSERT INTO Ticket (Issue, DateIssued, DateResolved,
      UserID, StaffID, StaffAssignedID, TicketCategoryID)
255 VALUES ('Mouse does not function', '2020-02-23 02:00:00',
      NULL, 1, NULL, 1, 701),
256      ('Applications keep crashing', '2020-01-02 04:00:00
      ', '2020-01-12 02:00:00', NULL, 2, 2, 702),
257      ('Applications are very slow', '2020-03-01 02:30:00
      ', '2020-03-07 02:00:00', NULL, 3, 3, 702),
258      ('Monitor is cracked', '2020-01-24 05:30:00', NULL
      , NULL, 4, NULL, 701),
259      ('User forgot password and security questions', '
      2020-01-03 05:00:00', '2020-01-04 02:00:00', 3, NULL, 4,
      703),
260      ('Internet Connection is down, says connected but
      no internet', '2020-04-02 07:00:00', NULL, 5, NULL, 5, 704
      ),
261      ('User forgot password', '2020-03-30 02:00:00', '

```

```

261 2020-04-10 05:30:00', 6, NULL, 6, 703),
262      ('Monitor fell and screen is cracked', '2020-02-01
263      01:00:00', NULL, 2, NULL, 4, 701),
264      ('User is confused how to change desktop background
265      and customizing the desktop', '2020-04-15 02:00:00', '
266      2020-04-16 08:00:00', 1, NULL, 2, 705),
267      ('User has questions about computer set up', '2020-
268      03-20 02:10:00', '2020-03-21 04:00:00', 2, NULL, 1, 705);
269
270 INSERT INTO Task (TaskName, TaskInformation,
271      DateTaskFinished, TicketID, StaffID)
272 VALUES ('Check hardware', 'Check the user mouse', '2020-02
273      -25 02:00:00', 1, 1),
274      ('Replace hardware', 'Order mouse for user and
275      replace it', NULL, 1, NULL),
276      ('Check software', 'Check user software', '2020-01-
277      07 04:00:00', 2, 2),
278      ('Fix software', 'Uninstall and reinstall software'
279      , '2020-01-12 02:00:00', 2, 2),
280      ('Check Software', 'Check user software', '2020-03-
281      04 04:00:00', 3, 3),
282      ('Fix Software', 'Update software', '2020-03-07 02:
283      00:00', 3, 3),
284      ('Replace hardware', 'Order monitor for user and
285      replace it', NULL, 4, NULL),
286      ('Reset account', 'reset user account', '2020-01-04
287      04:00:00', 5, 4),
288      ('Check connection', 'Check internet connection',
289      NULL, 6, 5),
290      ('Reset account', 'reset user account', '2020-04-10
291      05:00:00', 7, 6),
292      ('Replace hardware', 'Order monitor for user and
293      replace it', NULL, 8, 4),
294      ('Contact user', 'Send comment to user to get info
295      and help,', '2020-04-16 01:00:00', 9, 2),
296      ('Contact user', 'Send comment to user to get info
297      and help,', '2020-03-21 02:00:00', 10, 1);
298
299 INSERT INTO Comment (CommentDate, CommentMade, TicketID,
300      UserID, StaffID)
301 VALUES ('2020-03-02 02:00:00', 'What color do you want for
302      the mouse?', 1, NULL, 1),
303      ('2020-03-02 02:00:00', 'Pink please, or violet', 1
304      , 1, NULL),
305      ('2020-01-04 03:00:00', 'I forgot my security
306      questions', 5, 3, NULL),

```



```

285      ('2020-04-04 04:00:00', 'Did you try to unplug then
      plug back the modem?', 6, NULL, 5),
286      ('2020-04-16 05:00:00', 'How do I change or
      customize my desktop?', 9, 1, NULL),
287      ('2020-03-21 07:00:00', 'What do you need help with
      ?', 10, NULL, 1);
288
289 INSERT INTO HardwareTicket (TicketID, HardwareID)
290 VALUES (1, 7),
291          (4, 2),
292          (8, 2);
293
294
295 -- SELECT STATEMENTS
296 -- Display a list of all tickets submitted within a given
      month of the current year.
297 -- The month will be supplied to the routine as a word (e.
      g. April)
298 DROP PROCEDURE IF EXISTS usp_GetAprilTickets;
299 GO
300 CREATE PROCEDURE usp_GetAprilTickets
301     @month VARCHAR(32) , @year INT OUTPUT
302 AS
303 BEGIN
304     SELECT *, DATENAME(month, DateIssued) AS AprilTickets
305     FROM Ticket
306     WHERE DATENAME(month, DateIssued) LIKE @month
307     AND DATEPART(year, DateIssued) LIKE @year;
308 END;
309 GO
310
311 EXEC usp_GetAprilTickets April , 2020 ;
312 GO
313
314 -- Display a list of the top 10 tickets that have had the
      most activity in the form
315 -- of comments. A start date and end date will be supplied
      to the routine in the
316 -- form 'yyyy-mm-dd'.
317 DROP PROCEDURE IF EXISTS usp_GetMostCommentedTickets;
318 GO
319 CREATE PROCEDURE usp_GetMostCommentedTickets
320 @startDate DATETIME2, @endDate DATETIME2
321 AS BEGIN
322     SELECT TOP 10 t.TicketID, Issue, DateIssued,
323                  DateResolved, COUNT(c.TicketID) AS

```

```

323 CountComments
324     FROM Comment c
325     INNER JOIN Ticket t ON c.TicketID = t.TicketID
326     WHERE @startDate < DateIssued AND @endDate >
        DateIssued AND DateResolved IS NULL
327     GROUP BY t.TicketID, Issue, DateIssued, DateResolved
328     ORDER BY COUNT(c.TicketID) DESC;
329 END;
330 GO
331
332 EXEC usp_GetMostCommentedTickets '2019-04-02', '2020-04-26
    ';
333 GO
334
335 -- Display a list of tickets for a particular category,
    ordered by descending date,
336 -- with the corresponding date displayed in the format (
    Month dddd, yyyy) e.g.
337 -- November 21st, 2009. A category name will be supplied
    to the routine.
338 DROP PROCEDURE IF EXISTS
    usp_DisplayTicketsForSpecifiedCategory;
339 GO
340 CREATE PROCEDURE usp_DisplayTicketsForSpecifiedCategory
341 @TicketCategory VARCHAR(32)
342 AS
343 BEGIN
344     SELECT Issue, DATENAME(MM, DateIssued) + RIGHT(CONVERT
        (VARCHAR(12), DateIssued, 107), 9)
345     AS [Month DDDD, YYYY], Issue, UserID TicketCategoryID
    FROM Ticket
346     WHERE TicketCategoryID = (SELECT TicketCategoryID FROM
        TicketCategory
347                                WHERE CategoryName=@
        TicketCategory) ORDER BY DateIssued DESC;
348 END;
349 GO
350
351 EXEC usp_DisplayTicketsForSpecifiedCategory Hardware
352
353 -- Display a "page" of ticket information by passing, to
    the routine, a page
354 -- number and the number of tickets per page. For example
    , passing "1,10" will
355 -- return the first ten tickets (ordered by ticket id),
    but passing "2,10" will

```

```
356 -- return next ten tickets (i.e. page 2).
357 DROP PROCEDURE IF EXISTS usp_TicketPage;
358 GO
359 CREATE PROCEDURE usp_TicketPage
360     @PageNumber INT,
361     @TicketsPerPage INT
362 AS
363     BEGIN
364         SELECT * FROM Ticket ORDER BY TicketID
365             OFFSET ((@PageNumber-1)*@TicketsPerPage) ROWS
366             FETCH NEXT @TicketsPerPage ROWS ONLY
367     END;
368 GO
369
370 EXEC usp_TicketPage 1,5;
```