1. *Insert 100 rows into the following table:*

*CREATE TABLE TEST1 (TEST\_ID INT IDENTITY(1,1));*

**DECLARE @counter INT  
SET IDENTITY\_INSERT TEST1 ON  
SET @counter = 1**

**WHILE @counter <= 100  
BEGIN**

**INSERT INTO TEST1 (TEST\_ID) VALUES (@counter)  
SET @counter = @counter + 1**

**END**

**GO**

1. *Write SELECT query to list latest record for each [Id]:*

*CREATE TABLE TEST2 ([Id] VARCHAR(10), [Data] VARCHAR(250), [Date] SMALLDATETIME);*

*INSERT INTO TEST2 VALUES*

*('A', 'abc', '2016-10-01'),*

*('A', 'def', '2015-05-20'),*

*('B', 'xyz', '2014-05-20'),*

*('B', 'uvw', '2016-10-01'),*

*('B', 'rst', '2015-10-01');*

**SELECT T1.Id, T1.Data, T1.Date**

**FROM TEST2 T1**

**INNER JOIN**

**(SELECT Id AS LatestId, MAX(Date) AS LatestDate FROM TEST2 GROUP BY Id) T2**

**ON T1.Id = T2.LatestId AND T1.Date = T2.LatestDate**

1. *Write a single SELECT statement to list all unique values of attribute “z”:*

*CREATE TABLE TEST3 ([Data] XML);*

*INSERT INTO TEST3 VALUES*

*('<r><a z="1" /><a z="2" /></r>'),*

*('<r><b z="2" /><b z="3" /></r>'),*

*('<r><c z="3"><c z="4" /></c></r>');*

**SELECT DISTINCT col.value('@z','INT') AS ZValue**

**FROM TEST3 CROSS APPLY [DATA].nodes('/r//\*') AS XMLTable(col)**

1. *Consider a table with an [Id] column as well as BIT columns named [A] to [E] and some initial values:*

*CREATE TABLE TEST5 ([Id] INT, [A] BIT, [B] BIT, [C] BIT, [D] BIT, [E] BIT);*

*INSERT INTO TEST5 ([Id], [A], [C], [E]) VALUES (1, 'true', 'false', 'true');*

*INSERT INTO TEST5 ([Id], [A], [B], [C]) VALUES (2, 'true', 'true', 'true');*

*INSERT INTO TEST5 ([Id], [C], [D], [E]) VALUES (1, 'false', 'false', 'true');*

*Create a procedure which will accept parameters ([Id] INT, [Column] CHAR (1)) and*

*then invert the value in table #TEST5 for row matched by [Id] and column named*

*[Column]. No change is made if existing value is NULL.*

**CREATE PROCEDURE Test5Procedure @Id INT, @Column CHAR(1)**

**AS**

**IF (@Column = 'A')  
BEGIN**

**UPDATE TEST5**

**SET A = CASE**

**WHEN A = 'true' THEN 'false'**

**WHEN A = 'false' THEN 'true'**

**END**

**WHERE Id = @Id**

**END**

**IF (@Column = 'B')**

**BEGIN**

**UPDATE TEST5**

**SET B = CASE**

**WHEN B = 'true' THEN 'false'**

**WHEN B = 'false' THEN 'true'**

**END**

**WHERE Id = @Id**

**END**

**IF (@Column = 'C')**

**BEGIN**

**UPDATE TEST5**

**SET C = CASE**

**WHEN C = 'true' THEN 'false'**

**WHEN C = 'false' THEN 'true'**

**END**

**WHERE Id = @Id**

**END**

**IF (@Column = 'D')**

**BEGIN**

**UPDATE TEST5**

**SET D = CASE**

**WHEN D = 'true' THEN 'false'**

**WHEN D = 'false' THEN 'true'**

**END**

**WHERE Id = @Id**

**END**

**IF (@Column = 'E')**

**BEGIN**

**UPDATE TEST5**

**SET E = CASE**

**WHEN E = 'true' THEN 'false'**

**WHEN E = 'false' THEN 'true'**

**END**

**WHERE Id = @Id**

**END**

**GO**

**EXEC Test5Procedure @id = 1, @Column = "C";**

1. *Programmatically find the least number of hops needed to get from Id=1 to Id=15 using the grid provided below. For example, to get from Id=1 to Id=4 will require 2 hops - (1,3) followed by (3,4).*

*CREATE TABLE TEST6 ([FromId] INT, [ToId] INT);*

*INSERT INTO TEST6 VALUES (1,2), (1,3), (3,4), (4,5), (5,6), (5,7), (1,7), (2,8), (8,9), (9,11), (9,10), (7,10), (10,12), (10,14), (12,13), (14,15);*

**WITH recursion(FromId, ToId)**

**AS (**

**SELECT FromId, ToId**

**FROM TEST6**

**WHERE TEST6.ToId = 4 -- To 4**

**UNION ALL**

**SELECT TEST6.FromId, TEST6.ToId**

**FROM TEST6 INNER JOIN recursion on TEST6.ToId = recursion.FromId**

**Where TEST6.FromId = 1 -- From 1**

**)**

**SELECT COUNT(\*) AS LeastNumberOfHops FROM recursion;**