ASSIGNMENT 9

1.

```
  create table Tutor
  (TutorID int  not null PRIMARY KEY,
   CertDate date,
   [Status] varchar(25));

CREATE TABLE Student
( StudentID int not null primary key,
  [Read] decimal(2,1));

CREATE TABLE MatchHistory
( MatchID int not null primary key,
  TutorID int not null,
  StudentID int not null,
  StartDate date,
  EndDate date,
CONSTRAINT FK_TutorID FOREIGN KEY(TutorID) REFERENCES Tutor (TutorID),
CONSTRAINT FK_StudentID FOREIGN KEY(StudentID) REFERENCES Student (StudentID)
);

CREATE TABLE TutorReport
( MatchID int  not null,
[Month] DATE not null,
[Hours] INT,
Lessons int,
CONSTRAINT PK_TutorReport PRIMARY KEY (MatchID,[Month]));
```

2.

```
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (100, '1/5/2008',
'Active');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (101, '1/5/2008', 'Temp
Stop');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (102, '1/5/2008',
'Dropped');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (103, '5/22/2008',
'Active');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (104, '5/22/2008',
'Active');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (105, '5/22/2008', 'Temp
Stop');
INSERT INTO Tutor (TutorID, CertDate, [Status]) values (106, '5/22/2008',
'Active');

INSERT INTO Student (StudentID,[Read]) Values (3000, 2.3);
INSERT INTO Student (StudentID,[Read]) Values (3001, 5.6);
INSERT INTO Student (StudentID,[Read]) Values (3002, 1.3);
INSERT INTO Student (StudentID,[Read]) Values (3003, 3.3);
INSERT INTO Student (StudentID,[Read]) Values (3004, 2.7);
INSERT INTO Student (StudentID,[Read]) Values (3005, 4.8);
INSERT INTO Student (StudentID,[Read]) Values (3006, 7.8);
```

```
INSERT INTO Student (StudentID,[Read]) Values (3007, 1.5);

INSERT INTO MatchHistory VALUES (1, 100, 3000, '1/10/2008', NULL);
INSERT INTO MatchHistory VALUES (2, 101, 3001, '1/15/2008', '5/15/2008');
INSERT INTO MatchHistory VALUES (3, 102, 3002, '2/10/2008', '3/01/2008');
INSERT INTO MatchHistory VALUES (4, 106, 3003, '5/28/2008', NULL);
INSERT INTO MatchHistory VALUES (5, 103, 3004, '6/1/2008', '6/15/2008');
INSERT INTO MatchHistory VALUES (6, 104, 3005, '6/1/2008', '6/28/2008');
INSERT INTO MatchHistory VALUES (7, 104, 3006, '6/1/2008', NULL);

INSERT INTO TutorReport VALUES (1, '6/30/2008', 8, 4);
INSERT INTO TutorReport VALUES (4, '6/30/2008', 8, 6);
INSERT INTO TutorReport VALUES (5, '6/30/2008', 4, 4);
INSERT INTO TutorReport VALUES (4, '7/31/2008', 10, 5);
INSERT INTO TutorReport VALUES (1, '7/31/2008', 4, 2);
```

3.

```
ALTER TABLE Student Add Math decimal(2,1);
```

4.

```
ALTER TABLE Tutor Add [Subject] varchar(25) CONSTRAINT CK_SUBJECT CHECK (
[Subject] IN ('Reading', 'Math','ESL') );
```

5. This can be done by adjusting the database design by allowing one tutor to teach multiple subjects. By adding this new requirement, the database should be able to track tutor who is interested to teach more than one subject.

6.

```
SELECT mh.MatchID, mh.EndDate, mh.TutorID
   FROM MatchHistory mh
   WHERE  (mh.EndDate is null or mh.EndDate>='6/30/2008')
              and NOT EXISTS
              ( SELECT 1 FROM TutorReport tr
                WHERE mh.MatchID = tr.MatchID
                AND  tr.[Month] between '7/1/2008' and  '7/31/2008'
                )
   ORDER BY mh.MatchID;
```

7.

```
CREATE TABLE Person (
PersonID                    int not null
 Constraint PERPERSID_PK PRIMARY KEY,
LastName                    Varchar(25) not null,
FirstName                   Varchar(25) not null,
MiddleInit                  Varchar(1)  null,
StrAddress            Varchar(20),
City                     Varchar(20),
State                       Char(2),
Zip                               Varchar(10),
```

```
Phone                Varchar(14),
EMail             Varchar(25),
[Type]             Char(1));


INSERT INTO Person
VALUES ( 100, 'Rai','Arya', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732234', 'rai@gmail.com','S');
INSERT INTO Person
VALUES ( 101, 'Nano','Elite', NULL,'1163 Boylston Street', 'Boston', 'MA',
'02115',
             '6173732096', 'nano@gmail.com','S');
INSERT INTO Person
VALUES ( 102, 'Cai','Chen', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732000', 'cai@gmail.com','S');
INSERT INTO Person
VALUES ( 103, 'Rao','Sid', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173734560', 'sid@gmail.com','S');
INSERT INTO Person
VALUES ( 104, 'Nan','Liue', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '65737329700', 'lieu@gmail.com','S');
INSERT INTO Person
VALUES ( 105, 'Ziye','Ling', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732000', 'Ling@gmail.com','S');
INSERT INTO Person
VALUES ( 106, 'Bhatt','Pra', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732540', 'pra@gmail.com','S');


INSERT INTO Person
VALUES ( 3000, 'Jain','Kartik', NULL,'360 Huntington Ave', 'Boston', 'MA',
'02115',
             '6174732400', 'kartik@gmail.com','S');
INSERT INTO Person
VALUES ( 3001, 'Koli','Vivek', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732000', 'Koli@gmail.com','S');
INSERT INTO Person
VALUES ( 3002, 'Kori','Preeti', NULL,'360 Huntington Ave', 'Boston', 'MA',
'02115',
             '6173732000', 'Preeti@gmail.com','S');
INSERT INTO Person
VALUES ( 3003, 'Qutar','Omar', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732000', 'omar@gmail.com','S');
INSERT INTO Person
VALUES ( 3004, 'Luo','Yitian', NULL,'360 Huntington Ave', 'Boston', 'MA', '02115',
             '6173732000', 'Yitian@gmail.com','S');
INSERT INTO Person
VALUES ( 3005, 'Mora','Jeffrey', NULL,'360 Huntington Ave', 'Boston', 'MA',
'02115',
             '6173732000', 'Jeffrey@gmail.com','S');
INSERT INTO Person
VALUES ( 3006, 'Lemos','Stanny', NULL,'360 Huntington Ave', 'Boston', 'MA',
'02115',
             '6173732000', 'Lemos@gmail.com','S');
INSERT INTO Person
VALUES ( 3007, 'Pal','Pryanka', NULL,'360 Huntington Ave', 'Boston', 'MA',
'02115','6173732000', 'Pal@gmail.com','S');
```

8.

```
SELECT s.StudentID, mh.EndDate, p.LastName, SUM(tr.[Hours]) AS [Total Hours],
       SUM(tr.Lessons) AS [Total Lessons]
FROM Person p
        JOIN Student s ON p.PersonID = s.StudentID
        JOIN MatchHistory mh ON  s.StudentID = mh.StudentID
        LEFT JOIN TutorReport tr ON mh.MatchID = tr.MatchID
WHERE (mh.EndDate IS NULL OR mh.EndDate>'6/30/2008')
GROUP BY s.StudentID, mh.EndDate, p.LastName;
```

9.

```
SELECT Person.LastName, Person.FirstName, Tutor.[Status]
FROM Person JOIN Tutor ON
  Person.PersonID = Tutor.TutorID
WHERE Tutor.[Status]='Active';
```

10.

```
   SELECT p.FirstName, p.LastName, mh.MatchID, mh.EndDate, mh.TutorID
   FROM MatchHistory mh
   JOIN Person p on p.PersonID =mh.TutorID
   WHERE  (mh.EndDate is null  )-- active tutoring only
       and NOT EXISTS
       ( SELECT 1 FROM TutorReport tr
         WHERE mh.MatchID = tr.MatchID
         AND  tr.[Month] between '7/1/2008' and  '7/31/2008'
         )
   ORDER BY mh.MatchID;
```

11.

```
CREATE PROCEDURE AvailablbleTutors  AS
BEGIN
SELECT Person.LastName, Person.FirstName, Tutor.[Status]
FROM Person JOIN Tutor ON
  Person.PersonID = Tutor.TutorID
WHERE Tutor.[Status]='Active';
END
```

12.

```
CREATE FUNCTION IsTutorAvailable (@TutorID int)
returns char(1)
AS
BEGIN

RETURN
(
       SELECT CASE WHEN Status ='Active' THEN 'Y' ELSE 'N' END
       FROM Tutor WHERE TutorID = @TutorID
       )
END;
```