

## **DAMG6210 - Data Management and Database Design**

### ***Homework 08***

**1.**

```
CREATE DATABASE AdultLiteracy;
```

```
GO
```

```
USE AdultLiteracy;
```

```
GO
```

```
CREATE TABLE Tutor (
```

```
    TutorID INT PRIMARY KEY,
```

```
    CertDate DATE,
```

```
    Status VARCHAR(20)
```

```
);
```

```
CREATE TABLE Student (
```

```
    StudentID INT PRIMARY KEY,
```

```
    [Read] DECIMAL(3,1)
```

```
);
```

```
CREATE TABLE MatchHistory (
```

```
    MatchID INT PRIMARY KEY,
```

```
    TutorID INT,
```

```
    StudentID INT,
```

```
    StartDate DATE,
```

```
    EndDate DATE,
```

```
    CONSTRAINT FK_MatchHistory_Tutor FOREIGN KEY (TutorID) REFERENCES
```

```
Tutor(TutorID),
```

```
    CONSTRAINT FK_MatchHistory_Student FOREIGN KEY (StudentID) REFERENCES
```

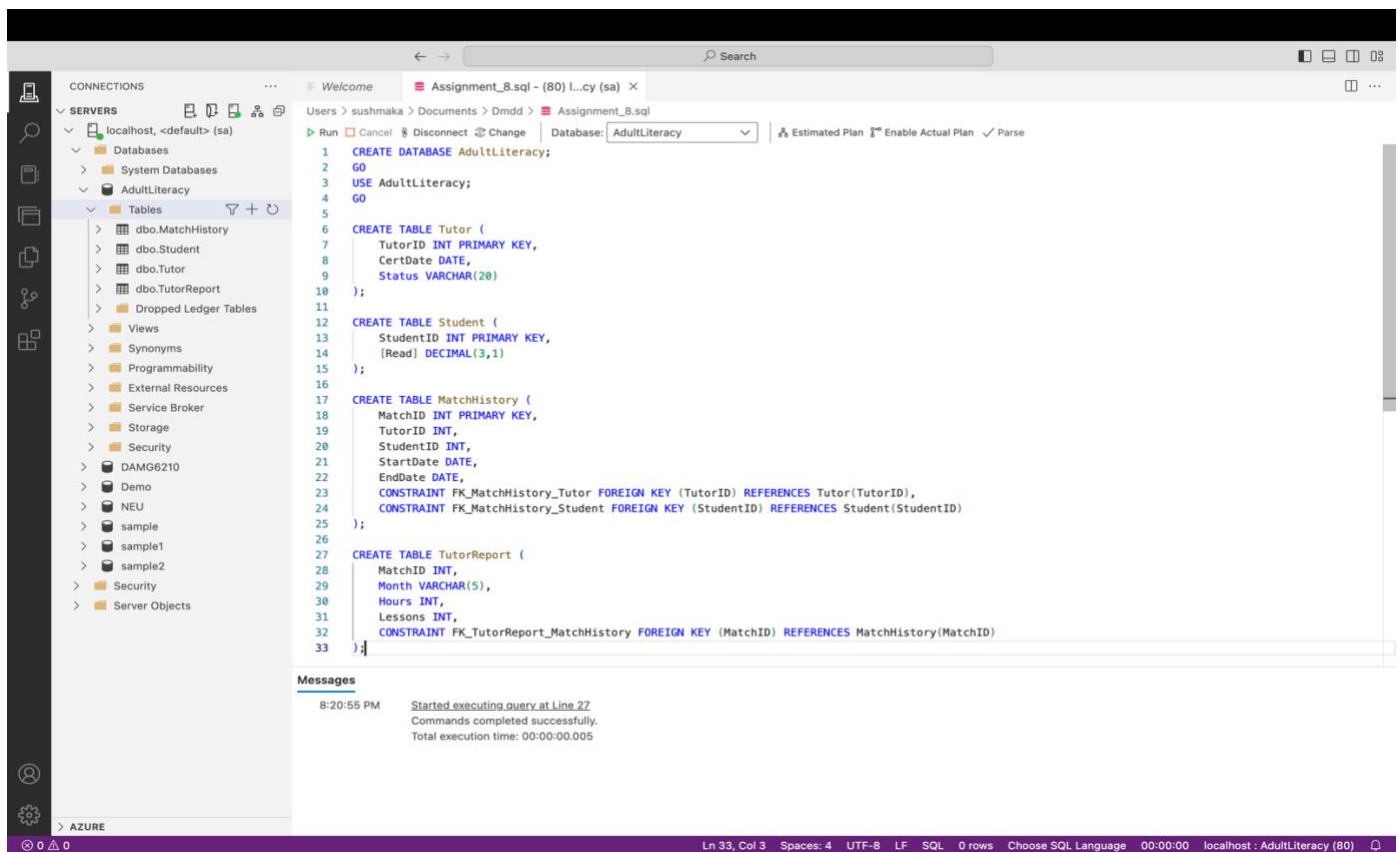
```
Student(StudentID)
```

```
);
```

```

CREATE TABLE TutorReport (
    MatchID INT,
    Month VARCHAR(5),
    Hours INT,
    Lessons INT,
    CONSTRAINT FK_TutorReport_MatchHistory FOREIGN KEY (MatchID) REFERENCES
MatchHistory(MatchID)
);

```



2.

```

INSERT INTO Tutor (TutorID, CertDate, Status) VALUES
(100, '2022-01-05', 'Active'),
(101, '2022-01-05', 'Temp_Stop'),
(102, '2022-01-05', 'Dropped'),

```

```
(103, '2022-05-22', 'Active'),  
(104, '2022-05-22', 'Active'),  
(105, '2022-05-22', 'Temp_Stop'),  
(106, '2022-05-22', 'Active');
```

```
-- Student
```

```
INSERT INTO Student (StudentID, [Read]) VALUES  
(3000, 2.3),  
(3001, 5.6),  
(3002, 1.3),  
(3003, 3.3),  
(3004, 2.7),  
(3005, 4.8),  
(3006, 7.8),  
(3007, 1.5);
```

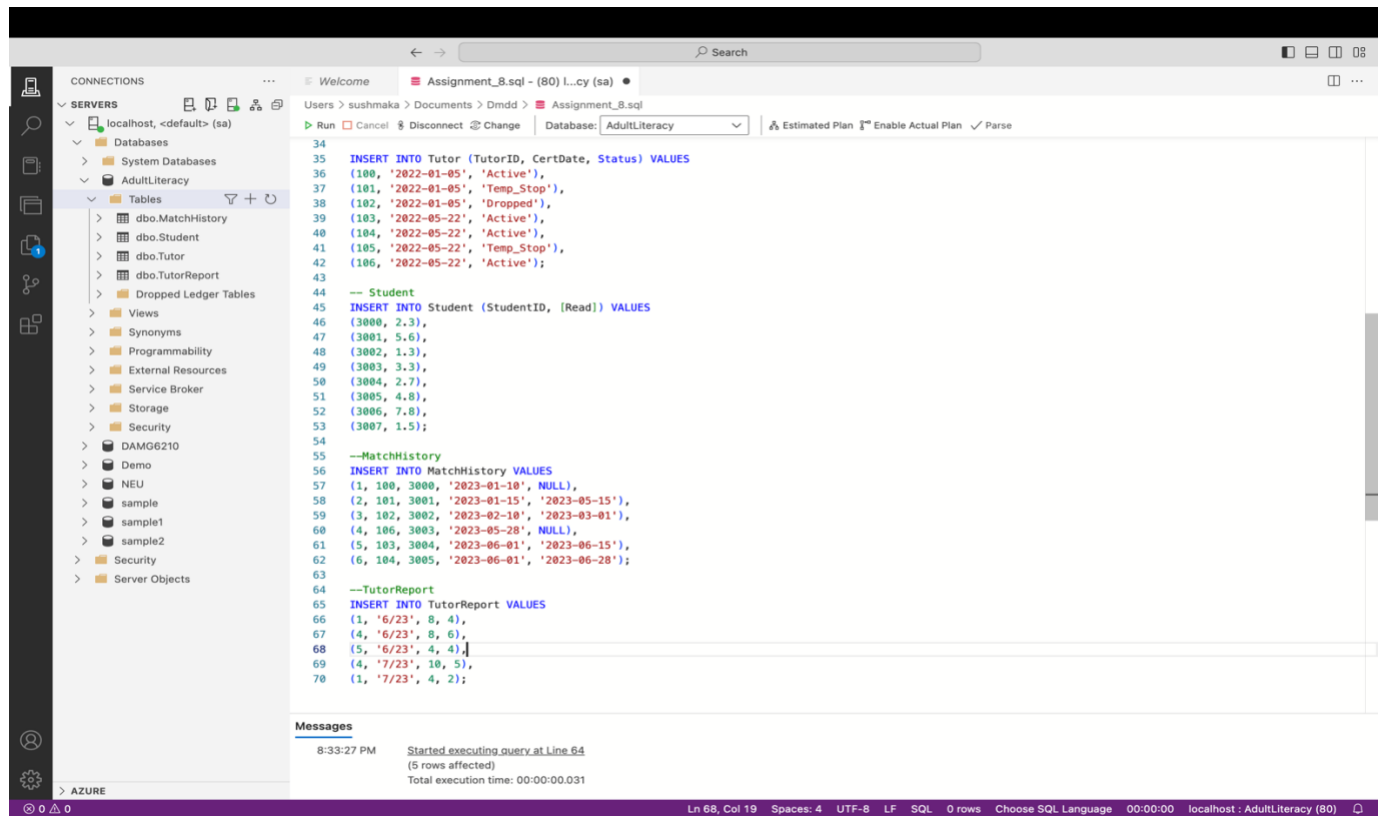
```
--MatchHistory
```

```
INSERT INTO MatchHistory VALUES  
(1, 100, 3000, '2023-01-10', NULL),  
(2, 101, 3001, '2023-01-15', '2023-05-15'),  
(3, 102, 3002, '2023-02-10', '2023-03-01'),  
(4, 106, 3003, '2023-05-28', NULL),  
(5, 103, 3004, '2023-06-01', '2023-06-15'),  
(6, 104, 3005, '2023-06-01', '2023-06-28');
```

```
--TutorReport
```

```
INSERT INTO TutorReport VALUES  
(1, '6/23', 8, 4),  
(4, '6/23', 8, 6),  
(5, '6/23', 4, 4),  
(4, '7/23', 10, 5),
```

(1, '7/23', 4, 2);



3.

ALTER TABLE Student ADD MathScore DECIMAL(3, 1);

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree is expanded to 'AdultLiteracy' > 'Tables'. The central query window displays the following SQL code:

```

61 (5, 103, 3004, '2023-06-01', '2023-06-15');
62 (6, 104, 3005, '2023-06-01', '2023-06-28');
63
64 --TutorReport
65 INSERT INTO TutorReport VALUES
66 (1, '6/23', 8, 4),
67 (4, '6/23', 8, 6),
68 (5, '6/23', 4, 4),
69 (4, '7/23', 10, 5),
70 (1, '7/23', 4, 2);
71
72
73
74
75 ALTER TABLE Student ADD MathScore DECIMAL(3, 1);
76
77 select * from student;
78
79

```

Below the code, the 'Results' tab shows the output of the query:

	StudentID	Read	MathScore
1	3000	2.3	NULL
2	3001	5.6	NULL
3	3002	1.3	NULL
4	3003	3.3	NULL
5	3004	2.7	NULL
6	3005	4.8	NULL
7	3006	7.8	NULL
8	3007	1.5	NULL

The status bar at the bottom indicates: Ln 73, Col 1 Spaces: 4 UTF-8 LF SQL 8 rows MSSQL 00:00:00 localhost : AdultLiteracy (80)

4.

ALTER TABLE Tutor

ADD Subject VARCHAR(10)

CONSTRAINT chk\_subject CHECK (Subject IN ('Reading', 'Math', 'ESL'));

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree is expanded to 'Tables' under the 'AdultLiteracy' database. The central query window displays the following SQL code:

```

81
82
83
84
85 ALTER TABLE Tutor
86 ADD Subject VARCHAR(10)
87 CONSTRAINT chk_subject CHECK (Subject IN ('Reading', 'Math', 'ESL'));
88
89 INSERT INTO Tutor (TutorID, CertDate, Status, Subject) VALUES
90 (107, '2022-08-01', 'Active', 'Reading');
91
92 SELECT * FROM Tutor;
93
94 DELETE FROM Tutor WHERE TutorID = 107;
95
96
97
98
99

```

Below the query window, the 'Results' tab is active, displaying a table with 8 rows and 4 columns: TutorID, CertDate, Status, and Subject.

	TutorID	CertDate	Status	Subject
1	100	2022-01-05	Active	NULL
2	101	2022-01-05	Temp_Stop	NULL
3	102	2022-01-05	Dropped	NULL
4	103	2022-05-22	Active	NULL
5	104	2022-05-22	Active	NULL
6	105	2022-05-22	Temp_Stop	NULL
7	106	2022-05-22	Active	NULL
8	107	2022-08-01	Active	Reading

The status bar at the bottom indicates: Ln 92, Col 1 (20 selected) Spaces: 4 UTF-8 LF SQL 8 rows MSSQL 00:00:00 localhost : AdultLiteracy (80)

5.

```

SELECT s.StudentID, m.TutorID, s.[Read]
FROM Student s
JOIN MatchHistory m ON s.StudentID = m.StudentID;

```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy for 'localhost, <default> (sa)'. The central pane shows the SQL script 'Assignment\_8.sql' with the following code:

```

68 (5, '6/23', 4, 4),
69 (4, '7/23', 10, 5),
70 (1, '7/23', 4, 2);
71
72 ALTER TABLE Student ADD MathScore DECIMAL(3, 1);
73
74 ALTER TABLE Tutor
75 ADD Subject VARCHAR(10)
76 CONSTRAINT chk_subject CHECK (Subject IN ('Reading', 'Math', 'ESL'));
77
78 SELECT s.StudentID, m.TutorID, s.[Read]
79 FROM Student s
80 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
81
82
83

```

The bottom pane shows the 'Results' tab with the following data:

	StudentID	TutorID	Read
1	3000	100	2.3
2	3001	101	5.6
3	3002	102	1.3
4	3003	106	3.3
5	3004	103	2.7
6	3005	104	4.8

The status bar at the bottom indicates: 'Ln 78, Col 1 (104 selected) Spaces: 4 UTF-8 LF SQL 6 rows Choose SQL Language 00:00:00 localhost : AdultLiteracy (80)'.

6.

SELECT s.StudentID, s.[Read], m.TutorID

FROM Student s

LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree is expanded to 'Tables' under the 'AdultLiteracy' database. The central query window displays the following SQL code:

```

68 (5, '6/23', 4, 4),
69 (4, '7/23', 10, 5),
70 (1, '7/23', 4, 2);
71
72 ALTER TABLE Student ADD MathScore DECIMAL(3, 1);
73
74 ALTER TABLE Tutor
75 ADD Subject VARCHAR(10)
76 CONSTRAINT chk_subject CHECK (Subject IN ('Reading', 'Math', 'ESL'));
77
78 SELECT s.StudentID, m.TutorID, s.[Read]
79 FROM Student s
80 JOIN MatchHistory m ON s.StudentID = m.StudentID;
81
82 SELECT s.StudentID, s.[Read], m.TutorID
83 FROM Student s
84 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
85
86
87

```

Below the query window, the 'Results' tab shows the output of the last query. The results are as follows:

	StudentID	Read	TutorID
1	3000	2.3	100
2	3001	5.6	101
3	3002	1.3	102
4	3003	3.3	106
5	3004	2.7	103
6	3005	4.8	104
7	3006	7.8	NULL
8	3007	1.5	NULL

The status bar at the bottom indicates: Ln 82, Col 1 (109 selected) Spaces: 4 UTF-8 LF SQL 8 rows Choose SQL Language 00:00:00 localhost: AdultLiteracy (80).

7.

SELECT t.TutorID, t.CertDate, m.StudentID

FROM Tutor t

LEFT JOIN MatchHistory m ON t.TutorID = m.TutorID;



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy for 'localhost, <default> (sa)'. The central pane shows a query window with the following SQL code:

```

68 (5, '6/23', 4, 4),
69 (4, '7/23', 10, 5),
70 (1, '7/23', 4, 2);
71
72 ALTER TABLE Student ADD MathScore DECIMAL(3, 1);
73
74 ALTER TABLE Tutor
75 ADD Subject VARCHAR(10)
76 CONSTRAINT chk_subject CHECK (Subject IN ('Reading', 'Math', 'ESL'));
77
78 SELECT s.StudentID, m.TutorID, s.[Read]
79 FROM Student s
80 JOIN MatchHistory m ON s.StudentID = m.StudentID;
81
82 SELECT s.StudentID, s.[Read], m.TutorID
83 FROM Student s
84 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
85
86 SELECT t.TutorID, t.CertDate, m.StudentID
87 FROM Tutor t
88 LEFT JOIN MatchHistory m ON t.TutorID = m.TutorID;
89
90
91
92

```

The bottom pane shows the 'Results' tab with a grid of data:

	TutorID	CertDate	StudentID
1	100	2022-01-05	3000
2	101	2022-01-05	3001
3	102	2022-01-05	3002
4	103	2022-05-22	3004
5	104	2022-05-22	3005
6	105	2022-05-22	NULL
7	106	2022-05-22	3003

The status bar at the bottom indicates 'Ln 86, Col 1 (105 selected) Spaces: 4 UTF-8 LF SQL 7 rows Choose SQL Language 00:00:00 localhost : AdultLiteracy (80)'.

8.

```

SELECT s.StudentID, s.[Read], m.MatchID
FROM Student s
JOIN MatchHistory m ON s.StudentID = m.StudentID
WHERE s.[Read] < (SELECT AVG([Read]) FROM Student);

```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree is expanded to 'Tables' under the 'AdultLiteracy' database. The main pane displays a SQL query with line numbers 82 to 98. The query is as follows:

```
82 SELECT s.StudentID, s.[Read], m.TutorID
83 FROM Student s
84 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
85
86 SELECT t.TutorID, t.CertDate, m.StudentID
87 FROM Tutor t
88 LEFT JOIN MatchHistory m ON t.TutorID = m.TutorID;
89
90 SELECT s.StudentID, s.[Read], m.MatchID
91 FROM Student s
92 JOIN MatchHistory m ON s.StudentID = m.StudentID
93 WHERE s.[Read] < (SELECT AVG([Read]) FROM Student);
94
95
96
97
98
```

Below the query, the 'Results' tab is active, showing a table with 4 rows and 3 columns: StudentID, Read, and MatchID.

	StudentID	Read	MatchID
1	3000	2.3	1
2	3002	1.3	3
3	3003	3.3	4
4	3004	2.7	5

The status bar at the bottom indicates 'Ln 90, Col 1 (155 selected)' and '4 rows'.

9.

SELECT TutorID

FROM MatchHistory

GROUP BY TutorID

HAVING COUNT(StudentID) > 1;

SSMS interface showing a SQL query in the Query Editor window. The query is a complex join and aggregation query. The Results pane is empty.

```
82 SELECT s.StudentID, s.[Read], m.TutorID
83 FROM Student s
84 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
85
86 SELECT t.TutorID, t.CertDate, m.StudentID
87 FROM Tutor t
88 LEFT JOIN MatchHistory m ON t.TutorID = m.TutorID;
89
90 SELECT s.StudentID, s.[Read], m.MatchID
91 FROM Student s
92 JOIN MatchHistory m ON s.StudentID = m.StudentID
93 WHERE s.[Read] < (SELECT AVG([Read]) FROM Student);
94
95 SELECT TutorID
96 FROM MatchHistory
97 GROUP BY TutorID
98 HAVING COUNT(StudentID) > 1;
```

Results Messages

TutorID
---------

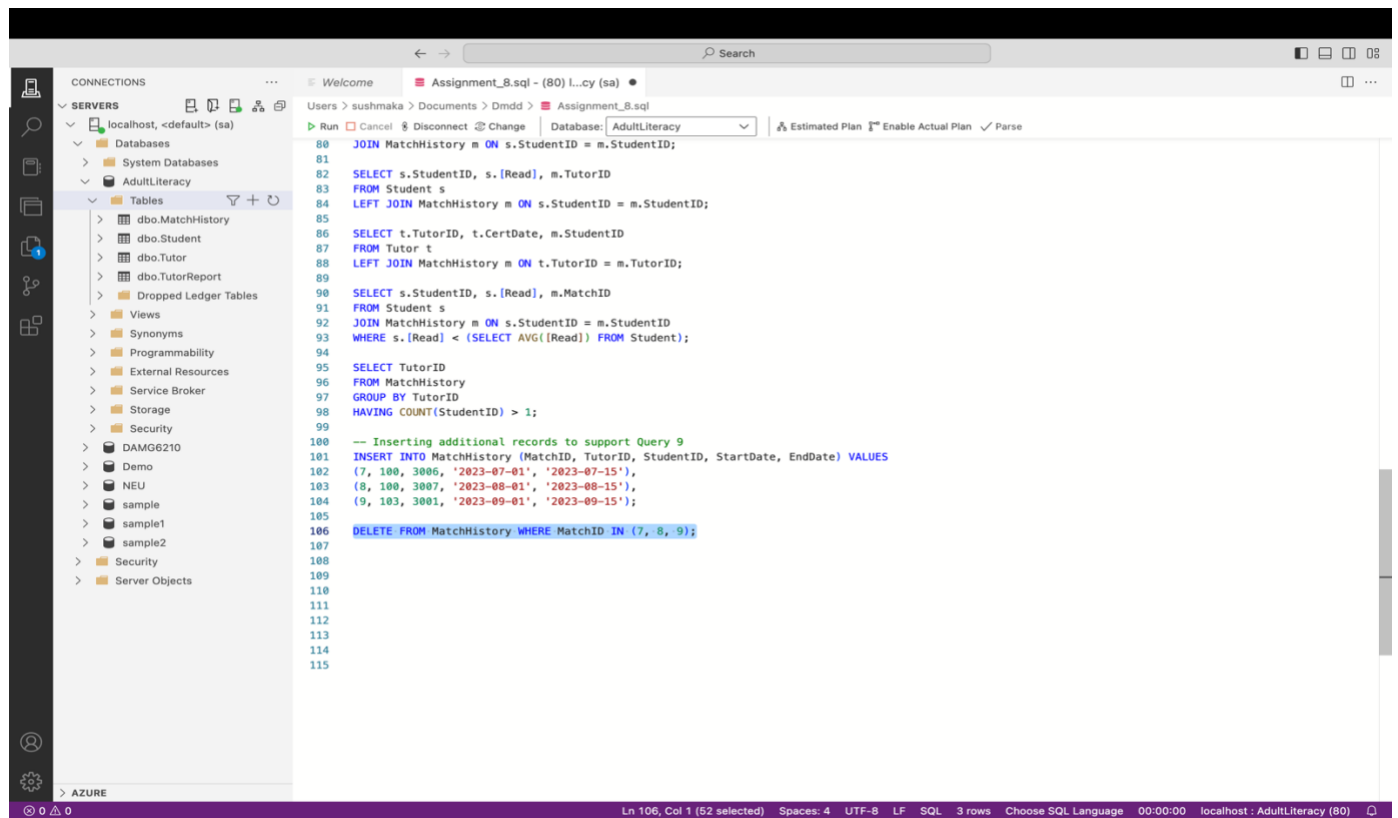
SSMS interface showing the same SQL query as above, but with an additional INSERT statement. The Results pane shows two rows of data.

```
82 SELECT s.StudentID, s.[Read], m.TutorID
83 FROM Student s
84 LEFT JOIN MatchHistory m ON s.StudentID = m.StudentID;
85
86 SELECT t.TutorID, t.CertDate, m.StudentID
87 FROM Tutor t
88 LEFT JOIN MatchHistory m ON t.TutorID = m.TutorID;
89
90 SELECT s.StudentID, s.[Read], m.MatchID
91 FROM Student s
92 JOIN MatchHistory m ON s.StudentID = m.StudentID
93 WHERE s.[Read] < (SELECT AVG([Read]) FROM Student);
94
95 SELECT TutorID
96 FROM MatchHistory
97 GROUP BY TutorID
98 HAVING COUNT(StudentID) > 1;
```

```
100 -- Inserting additional records to support Query 9
101 INSERT INTO MatchHistory (MatchID, TutorID, StudentID, StartDate, EndDate) VALUES
102 (7, 100, 3006, '2023-07-01', '2023-07-15'),
103 (8, 100, 3007, '2023-08-01', '2023-08-15'),
104 (9, 103, 3001, '2023-09-01', '2023-09-15');
```

Results Messages

TutorID
100
103



10.

```

SELECT m.TutorID, SUM(tr.Lessons) as TotalLessons
FROM MatchHistory m
INNER JOIN TutorReport tr ON m.MatchID = tr.MatchID
GROUP BY m.TutorID;

```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree is expanded to 'Tables' under the 'AdultLiteracy' database. The main window displays a SQL query script with line numbers 92 to 118. The query includes a JOIN, a subquery for average reads, an INSERT statement, and a DELETE statement. Below the script, the 'Results' tab is active, showing a table with 3 rows and 2 columns: TutorID and TotalLessons.

```

92 JOIN MatchHistory m ON s.StudentID = m.StudentID
93 WHERE s.[Read] < (SELECT AVG([Read]) FROM Student);
94
95 SELECT TutorID
96 FROM MatchHistory
97 GROUP BY TutorID
98 HAVING COUNT(StudentID) > 1;
99
100 -- Inserting additional records to support Query 9
101 INSERT INTO MatchHistory (MatchID, TutorID, StudentID, StartDate, EndDate) VALUES
102 (7, 100, 3006, '2023-07-01', '2023-07-15'),
103 (8, 100, 3007, '2023-08-01', '2023-08-15'),
104 (9, 103, 3001, '2023-09-01', '2023-09-15');
105
106 DELETE FROM MatchHistory WHERE MatchID IN (7, 8, 9);
107
108 SELECT m.TutorID, SUM(tr.Lessons) as TotalLessons
109 FROM MatchHistory m
110 INNER JOIN TutorReport tr ON m.MatchID = tr.MatchID
111 GROUP BY m.TutorID;
112
113
114
115
116
117
118

```

	TutorID	TotalLessons
1	100	6
2	103	4
3	106	11

11.

SELECT s.StudentID, m.TutorID, s.[Read]

FROM Student s

JOIN MatchHistory m ON s.StudentID = m.StudentID

WHERE s.[Read] > (

SELECT AVG(s2.[Read])

FROM Student s2

JOIN MatchHistory m2 ON s2.StudentID = m2.StudentID

WHERE m2.TutorID = m.TutorID

);

SSMS interface showing a SQL query in the query editor. The query is a complex join and aggregation query. The left pane shows the database structure, including tables like MatchHistory, Student, Tutor, and TutorReport. The right pane shows the query results.

```
109 FROM MatchHistory m
110 INNER JOIN TutorReport tr ON m.MatchID = tr.MatchID
111 GROUP BY m.TutorID;
112
113 SELECT s.StudentID, m.TutorID, s.[Read]
114 FROM Student s
115 JOIN MatchHistory m ON s.StudentID = m.StudentID
116 WHERE s.[Read] > (
117     SELECT AVG(s2.[Read])
118     FROM Student s2
119     JOIN MatchHistory m2 ON s2.StudentID = m2.StudentID
120     WHERE m2.TutorID = m.TutorID
121 );
```

Results

StudentID	TutorID	Read
3006	100	7.8
3001	103	5.6

Ln 113, Col 1 (260 selected) Spaces: 4 UTF-8 LF SQL 0 rows MSSQL 00:00:00 localhost: AdultLiteracy (80)

SSMS interface showing a SQL query in the query editor. The query is a complex join and aggregation query. The left pane shows the database structure, including tables like MatchHistory, Student, Tutor, and TutorReport. The right pane shows the query results.

```
112 -- Inserting additional records to support Query 9
113 INSERT INTO MatchHistory (MatchID, TutorID, StudentID, StartDate, EndDate) VALUES
114 (7, 100, 3006, '2023-07-01', '2023-07-15'),
115 (8, 100, 3007, '2023-08-01', '2023-08-15'),
116 (9, 103, 3001, '2023-09-01', '2023-09-15');
117
118 SELECT s.StudentID, m.TutorID, s.[Read]
119 FROM Student s
120 JOIN MatchHistory m ON s.StudentID = m.StudentID
121 WHERE s.[Read] > (
122     SELECT AVG(s2.[Read])
123     FROM Student s2
124     JOIN MatchHistory m2 ON s2.StudentID = m2.StudentID
125     WHERE m2.TutorID = m.TutorID
126 );
```

Results

StudentID	TutorID	Read	
1	3006	100	7.8
2	3001	103	5.6

Ln 113, Col 1 (529 selected) Spaces: 4 UTF-8 LF SQL 2 rows MSSQL 00:00:00 localhost: AdultLiteracy (80)

## **References**

Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management* (13th ed.). Pearson