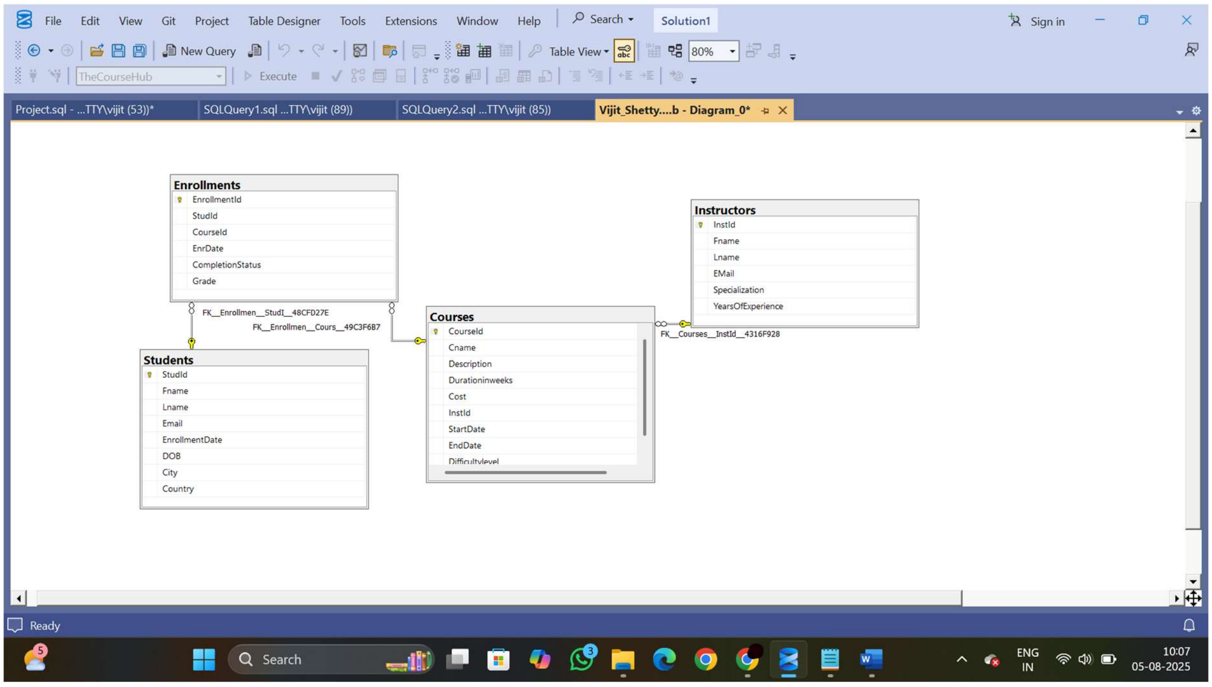
KA -25 Batch 2

SQL Project

Vijit Shetty

Title: Database design for TheCourseHub.

ER Diagram :

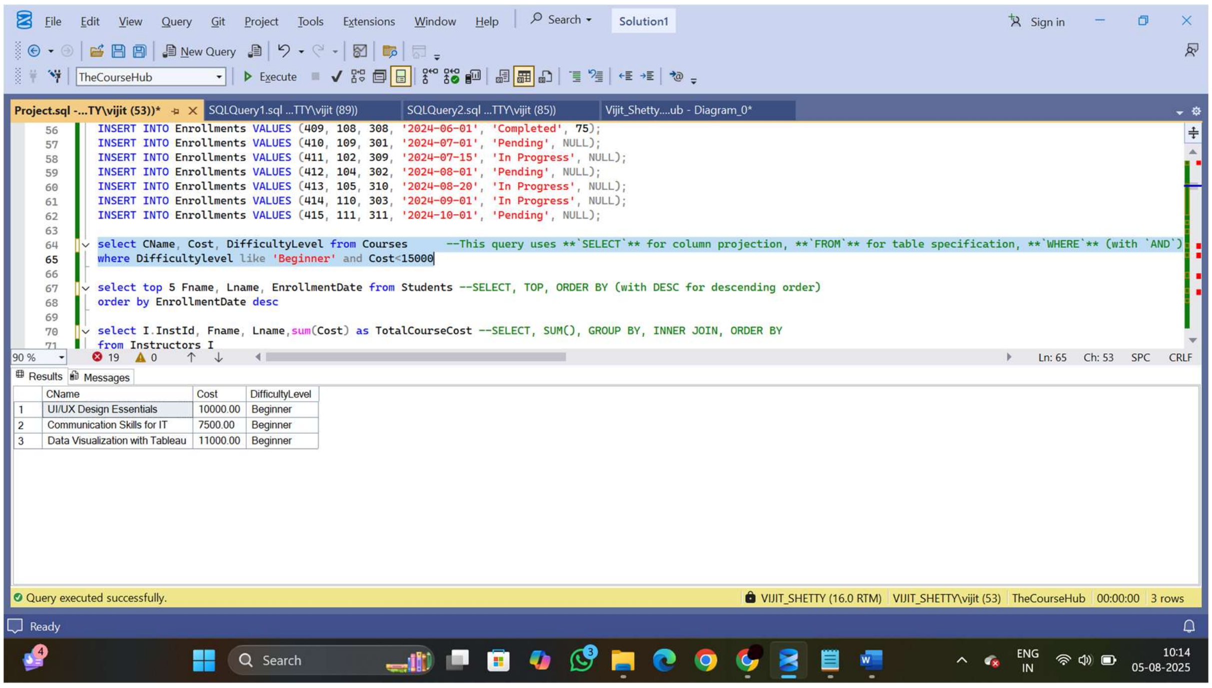


Following are the queries that include all the major concepts covered during the training.

1.Display the CourseName, Cost, and DifficultyLevel for all courses that are classified as 'Beginner' level and have a Cost less than ₹15,000. Sort the results by Cost in ascending order.

Concepts Used: `SELECT` for column projection, `FROM` for table specification, `WHERE` (with `AND`) for row filtering based on multiple conditions, and `ORDER BY` for sorting the result set.

Snapshot :



```
INSERT INTO Enrollments VALUES (409, 108, 308, '2024-06-01', 'Completed', 75);
INSERT INTO Enrollments VALUES (410, 109, 301, '2024-07-01', 'Pending', NULL);
INSERT INTO Enrollments VALUES (411, 102, 309, '2024-07-15', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (412, 104, 302, '2024-08-01', 'Pending', NULL);
INSERT INTO Enrollments VALUES (413, 105, 310, '2024-08-20', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (414, 110, 303, '2024-09-01', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (415, 111, 311, '2024-10-01', 'Pending', NULL);

select CName, Cost, DifficultyLevel from Courses      --This query uses **`SELECT`** for column projection, **`FROM`** for table specification, **`WHERE`** (with `AND`)
where Difficultylevel like 'Beginner' and Cost<15000

select top 5 Fname, Lname, EnrollmentDate from Students --SELECT, TOP, ORDER BY (with DESC for descending order)
order by EnrollmentDate desc

select I.InstId, Fname, Lname,sum(Cost) as TotalCourseCost --SELECT, SUM(), GROUP BY, INNER JOIN, ORDER BY
from Instructors I
```

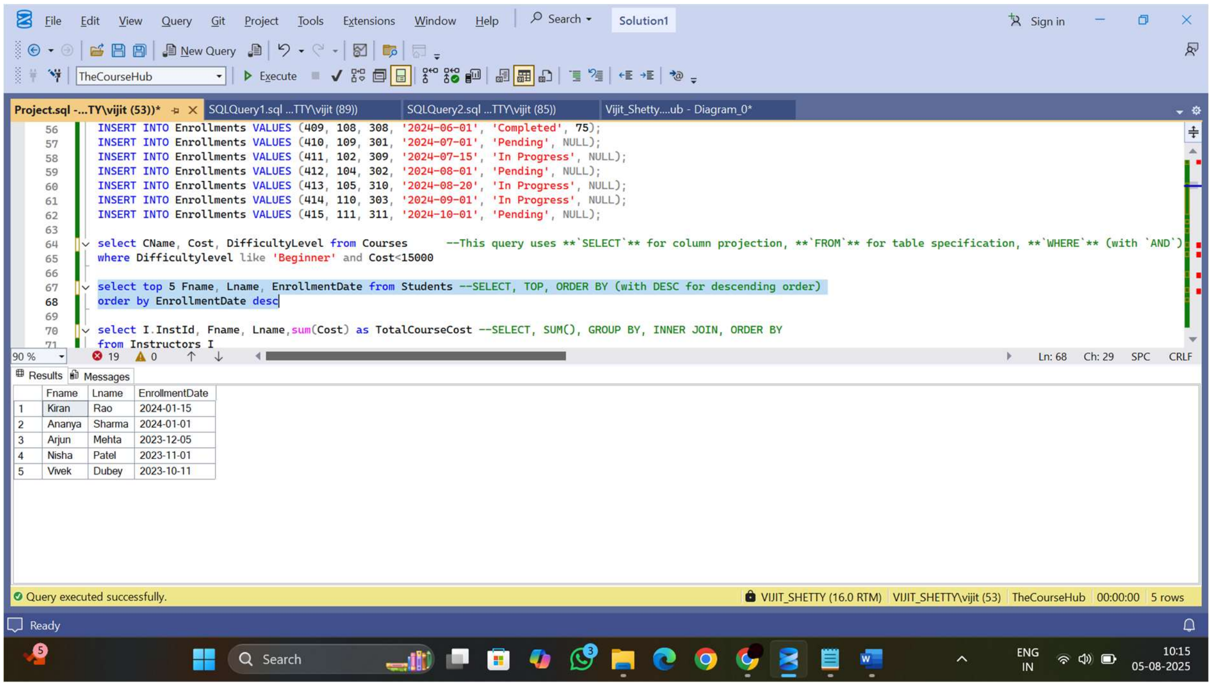
	CName	Cost	DifficultyLevel
1	UI/UX Design Essentials	10000.00	Beginner
2	Communication Skills for IT	7500.00	Beginner
3	Data Visualization with Tableau	11000.00	Beginner

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 3 rows

2.List the Fname, Lname, and EnrollmentDate of the 5 most recently enrolled students.

Concept Used: SELECT, TOP, ORDER BY (with DESC for descending order)

Snapshot:



```
INSERT INTO Enrollments VALUES (409, 108, 308, '2024-06-01', 'Completed', 75);
INSERT INTO Enrollments VALUES (410, 109, 301, '2024-07-01', 'Pending', NULL);
INSERT INTO Enrollments VALUES (411, 102, 309, '2024-07-15', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (412, 104, 302, '2024-08-01', 'Pending', NULL);
INSERT INTO Enrollments VALUES (413, 105, 310, '2024-08-20', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (414, 110, 303, '2024-09-01', 'In Progress', NULL);
INSERT INTO Enrollments VALUES (415, 111, 311, '2024-10-01', 'Pending', NULL);

select CName, Cost, DifficultyLevel from Courses      --This query uses **`SELECT`** for column projection, **`FROM`** for table specification, **`WHERE`** (with `AND`)
where Difficultylevel like 'Beginner' and Cost<15000

select top 5 Fname, Lname, EnrollmentDate from Students --SELECT, TOP, ORDER BY (with DESC for descending order)
order by EnrollmentDate desc

select I.InstId, Fname, Lname,sum(Cost) as TotalCourseCost --SELECT, SUM(), GROUP BY, INNER JOIN, ORDER BY
from Instructors I
```

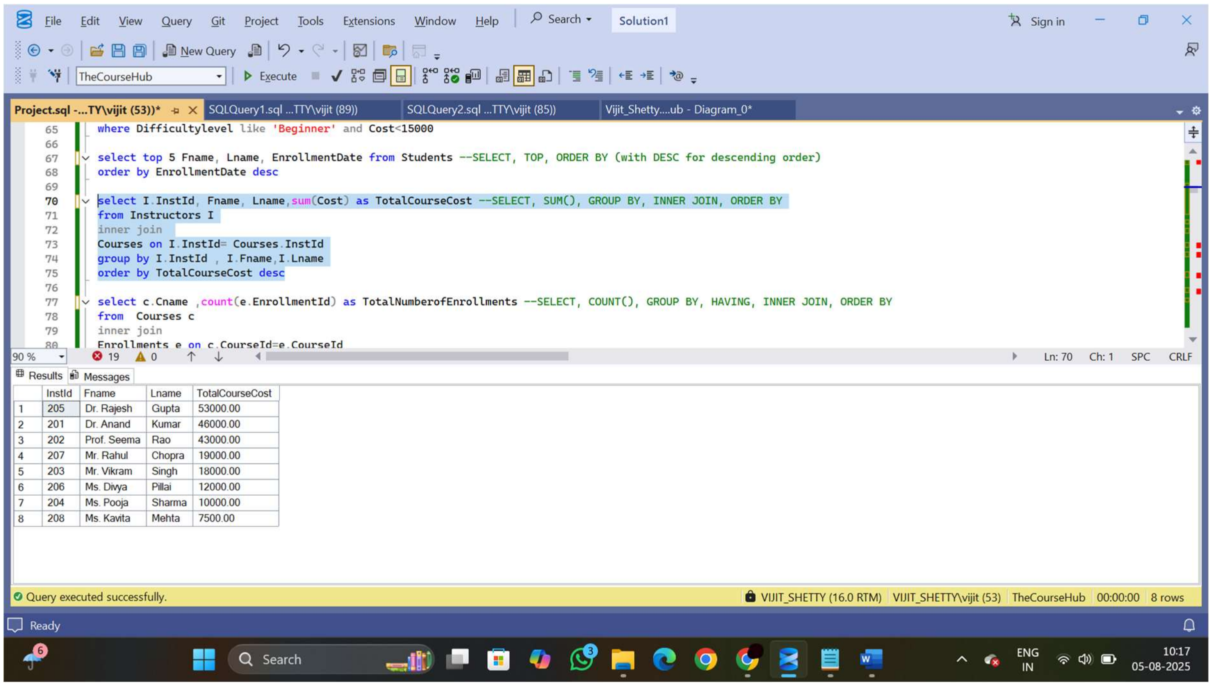
	Fname	Lname	EnrollmentDate
1	Kiran	Rao	2024-01-15
2	Ananya	Sharma	2024-01-01
3	Arjun	Mehta	2023-12-05
4	Nisha	Patel	2023-11-01
5	Vivek	Dubey	2023-10-11

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 5 rows

3.Calculate the total Cost of all courses taught by each instructor. Display the InstructorID, FirstName, LastName, and the TotalCourseCost.

Concept: SELECT, SUM(), GROUP BY, INNER JOIN, ORDER BY

Snapshot :



```
where Difficultylevel like 'Beginner' and Cost<15000

select top 5 Fname, Lname, EnrollmentDate from Students --SELECT, TOP, ORDER BY (with DESC for descending order)
order by EnrollmentDate desc

select I.InstId, Fname, Lname,sum(Cost) as TotalCourseCost --SELECT, SUM(), GROUP BY, INNER JOIN, ORDER BY
from Instructors I
inner join
Courses on I.InstId= Courses.InstId
group by I.InstId , I.Fname,I.Lname
order by TotalCourseCost desc

select c.Cname ,count(e.EnrollmentId) as TotalNumberofEnrollments --SELECT, COUNT(), GROUP BY, HAVING, INNER JOIN, ORDER BY
from  Courses c
inner join
Enrollments e on c.CourseId=e.CourseId
```

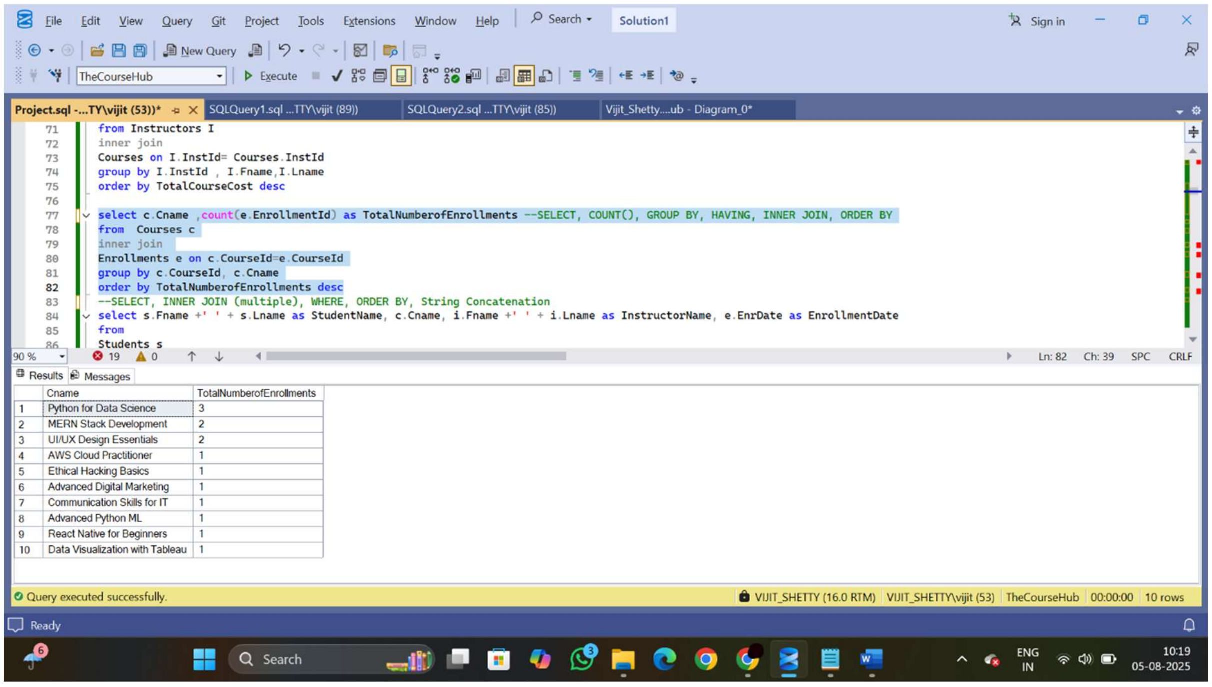
	InstId	Fname	Lname	TotalCourseCost
1	205	Dr. Rajesh	Gupta	53000.00
2	201	Dr. Anand	Kumar	46000.00
3	202	Prof. Seema	Rao	43000.00
4	207	Mr. Rahul	Chopra	19000.00
5	203	Mr. Vikram	Singh	18000.00
6	206	Ms. Divya	Pillai	12000.00
7	204	Ms. Pooja	Sharma	10000.00
8	208	Ms. Kavita	Mehta	7500.00

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 8 rows

4.Find the CourseName and the Number of Enrollments for courses that have been enrolled in more than 2 times. Sort the results by the number of enrollments in descending order.

Concept : SELECT, COUNT(), GROUP BY, HAVING, INNER JOIN, ORDER BY

Snapshot:



```
from Instructors I
inner join
Courses on I.InstId= Courses.InstId
group by I.InstId , I.Fname,I.Lname
order by TotalCourseCost desc

select c.Cname ,count(e.EnrollmentId) as TotalNumberofEnrollments --SELECT, COUNT(), GROUP BY, HAVING, INNER JOIN, ORDER BY
from  Courses c
inner join
Enrollments e on c.CourseId=e.CourseId
group by c.CourseId, c.Cname
order by TotalNumberofEnrollments desc
--SELECT, INNER JOIN (multiple), WHERE, ORDER BY, String Concatenation
select s.Fname +' ' + s.Lname as StudentName, c.Cname, i.Fname +' ' + i.Lname as InstructorName, e.EnrDate as EnrollmentDate
from
Students s
```

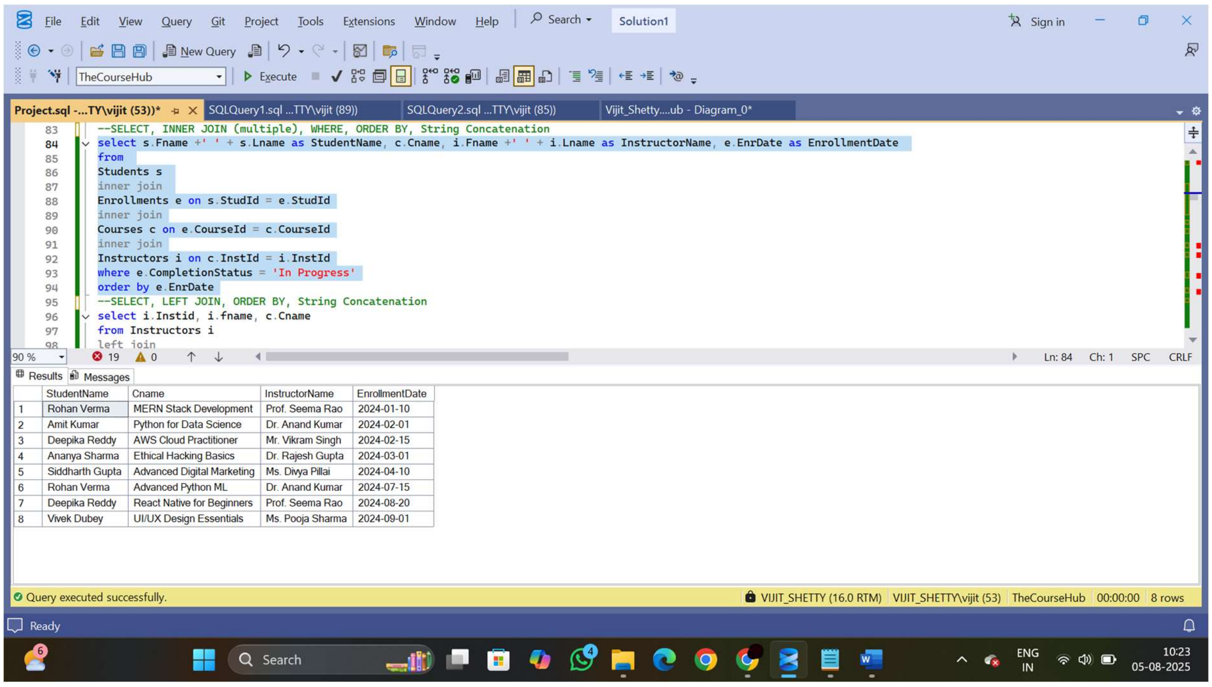
	Cname	TotalNumberofEnrollments
1	Python for Data Science	3
2	MERN Stack Development	2
3	UI/UX Design Essentials	2
4	AWS Cloud Practitioner	1
5	Ethical Hacking Basics	1
6	Advanced Digital Marketing	1
7	Communication Skills for IT	1
8	Advanced Python ML	1
9	React Native for Beginners	1
10	Data Visualization with Tableau	1

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 10 rows

5.Display the Student's Full Name, Course Name, Instructor's Full Name, and Enrollment Date for all enrollments that are currently 'In Progress'.

Concept : SELECT, INNER JOIN (multiple), WHERE, ORDER BY, String Concatenation

Snapshot:



```
--SELECT, INNER JOIN (multiple), WHERE, ORDER BY, String Concatenation
select s.Fname +' ' + s.Lname as StudentName, c.Cname, i.Fname +' ' + i.Lname as InstructorName, e.EnrDate as EnrollmentDate
from
Students s
inner join
Enrollments e on s.StudId = e.StudId
inner join
Courses c on e.CourseId = c.CourseId
inner join
Instructors i on c.InstId = i.InstId
where e.CompletionStatus = 'In Progress'
order by e.EnrDate
--SELECT, LEFT JOIN, ORDER BY, String Concatenation
select i.Instid, i.fname, c.Cname
from Instructors i
left join
```

	StudentName	Cname	InstructorName	EnrollmentDate
1	Rohan Verma	MERN Stack Development	Prof. Seema Rao	2024-01-10
2	Amit Kumar	Python for Data Science	Dr. Anand Kumar	2024-02-01
3	Deepika Reddy	AWS Cloud Practitioner	Mr. Vikram Singh	2024-02-15
4	Ananya Sharma	Ethical Hacking Basics	Dr. Rajesh Gupta	2024-03-01
5	Siddharth Gupta	Advanced Digital Marketing	Ms. Divya Pillai	2024-04-10
6	Rohan Verma	Advanced Python ML	Dr. Anand Kumar	2024-07-15
7	Deepika Reddy	React Native for Beginners	Prof. Seema Rao	2024-08-20
8	Vivek Dubey	UI/UX Design Essentials	Ms. Pooja Sharma	2024-09-01

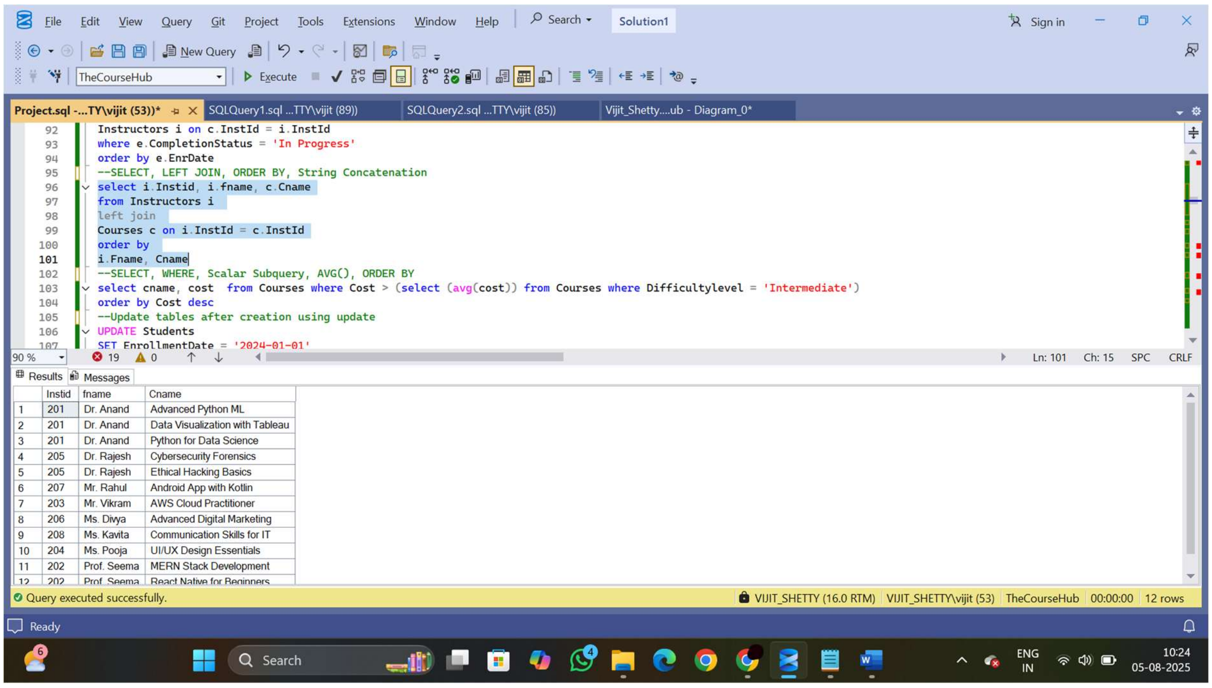
Query executed successfully. VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 8 rows

6.List all instructors and the names of the courses they teach. Include instructors who currently do not have any courses assigned to them.

Concept :

SELECT, LEFT JOIN, ORDER BY, String Concatenation

Snapshot:



```sql
Instructors i on c.InstId = i.InstId
where e.CompletionStatus = 'In Progress'
order by e.EnrDate
--SELECT, LEFT JOIN, ORDER BY, String Concatenation
select i.Instid, i.fname, c.Cname
from Instructors i
left join
Courses c on i.InstId = c.InstId
order by
i.Fname, Cname
--SELECT, WHERE, Scalar Subquery, AVG(), ORDER BY
select cname, cost  from Courses where Cost > (select (avg(cost)) from Courses where Difficultylevel = 'Intermediate')
order by Cost desc
--Update tables after creation using update
UPDATE Students
SET EnrollmentDate = '2024-01-01'
```

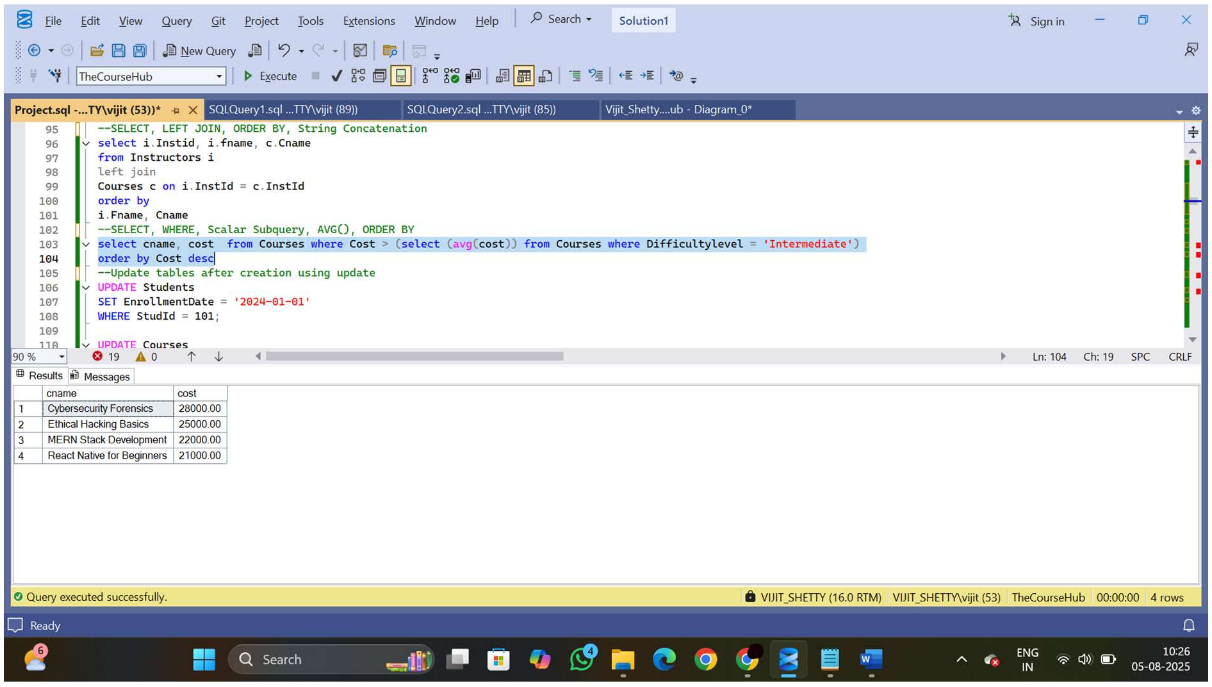
	Instid	fname	Cname
1	201	Dr. Anand	Advanced Python ML
2	201	Dr. Anand	Data Visualization with Tableau
3	201	Dr. Anand	Python for Data Science
4	205	Dr. Rajesh	Cybersecurity Forensics
5	205	Dr. Rajesh	Ethical Hacking Basics
6	207	Mr. Rahul	Android App with Kotlin
7	203	Mr. Vikram	AWS Cloud Practitioner
8	206	Ms. Divya	Advanced Digital Marketing
9	208	Ms. Kavita	Communication Skills for IT
10	204	Ms. Pooja	UI/UX Design Essentials
11	202	Prof. Seema	MERN Stack Development
12	202	Prof. Seema	React Native for Beginners

Query executed successfully. VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 12 rows

7.Find CourseName and Cost for all courses that are more expensive than the average cost of 'Intermediate' level courses.

Concept : SELECT, WHERE, Scalar Subquery, AVG(), ORDER BY

Snapshot:



```sql
--SELECT, LEFT JOIN, ORDER BY, String Concatenation
select i.Instid, i.fname, c.Cname
from Instructors i
left join
Courses c on i.InstId = c.InstId
order by
i.Fname, Cname
--SELECT, WHERE, Scalar Subquery, AVG(), ORDER BY
select cname, cost  from Courses where Cost > (select (avg(cost)) from Courses where Difficultylevel = 'Intermediate')
order by Cost desc
--Update tables after creation using update
UPDATE Students
SET EnrollmentDate = '2024-01-01'
WHERE StudId = 101;

UPDATE Courses
```

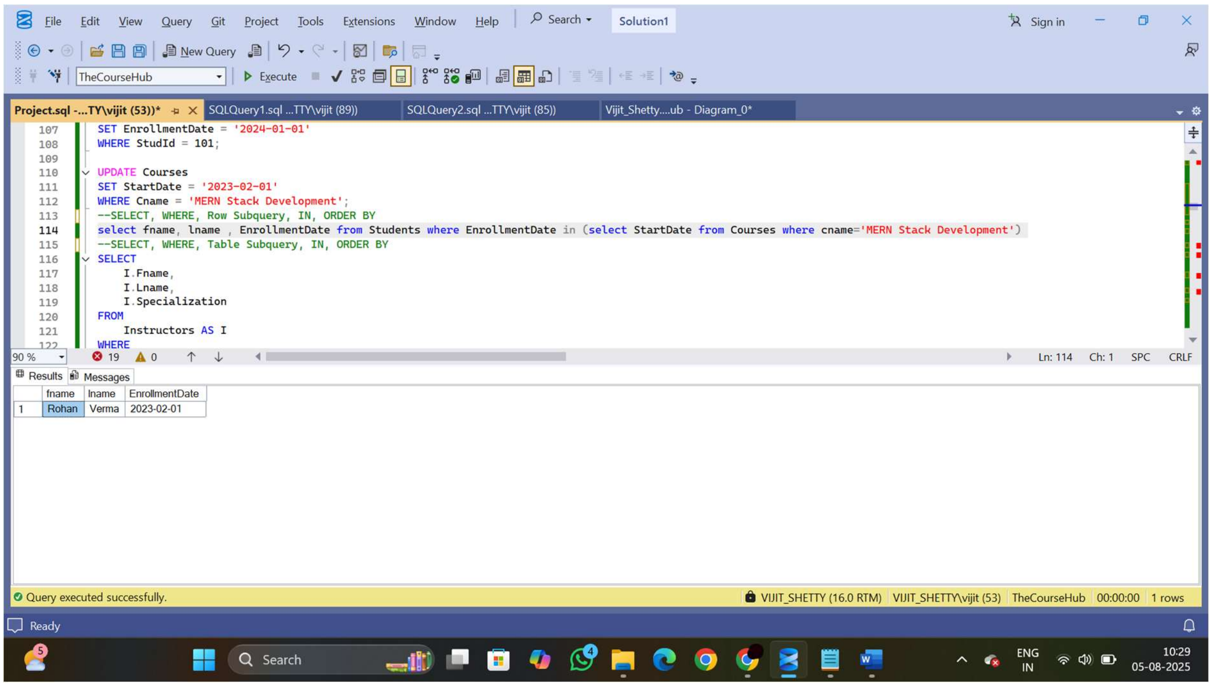
	cname	cost
1	Cybersecurity Forensics	28000.00
2	Ethical Hacking Basics	25000.00
3	MERN Stack Development	22000.00
4	React Native for Beginners	21000.00

Query executed successfully. VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 4 rows

8.List the FirstName, LastName, and EnrollmentDate of students who enrolled in any course on the same day that the 'MERN Stack Development' course started.

Concept : SELECT, WHERE, Row Subquery, IN, ORDER BY

Snapshot :



```sql
SET EnrollmentDate = '2024-01-01'
WHERE StudId = 101;

UPDATE Courses
SET StartDate = '2023-02-01'
WHERE Cname = 'MERN Stack Development';
--SELECT, WHERE, Row Subquery, IN, ORDER BY
select fname, lname , EnrollmentDate from Students where EnrollmentDate in (select StartDate from Courses where cname='MERN Stack Development')
--SELECT, WHERE, Table Subquery, IN, ORDER BY
SELECT
    I.Fname,
    I.Lname,
    I.Specialization
FROM
    Instructors AS I
WHERE
```

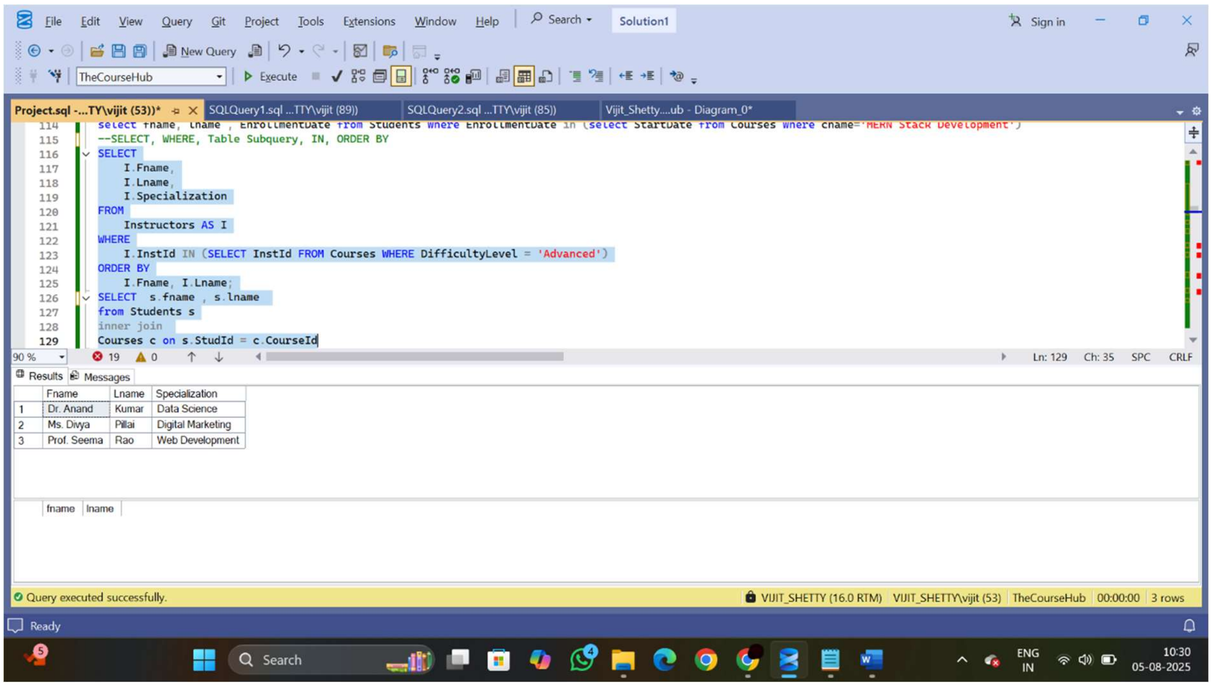
	fname	lname	EnrollmentDate
1	Rohan	Verma	2023-02-01

Query executed successfully. VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 1 rows

9.Display the FirstName, LastName, and Specialization of instructors who teach at least one 'Advanced' level course.

Concept : SELECT, WHERE, Table Subquery, IN, ORDER BY

Snapshot:



```sql
select fname, lname , EnrollmentDate from Students where EnrollmentDate in (select StartDate from Courses where cname='MERN Stack Development')
--SELECT, WHERE, Table Subquery, IN, ORDER BY
SELECT
    I.Fname,
    I.Lname,
    I.Specialization
FROM
    Instructors AS I
WHERE
    I.InstId IN (SELECT InstId FROM Courses WHERE DifficultyLevel = 'Advanced')
ORDER BY
    I.Fname, I.Lname;
SELECT  s.fname , s.lname
from Students s
inner join
Courses c on s.StudId = c.CourseId
```

	Fname	Lname	Specialization
1	Dr. Anand	Kumar	Data Science
2	Ms. Divya	Pillai	Digital Marketing
3	Prof. Seema	Rao	Web Development

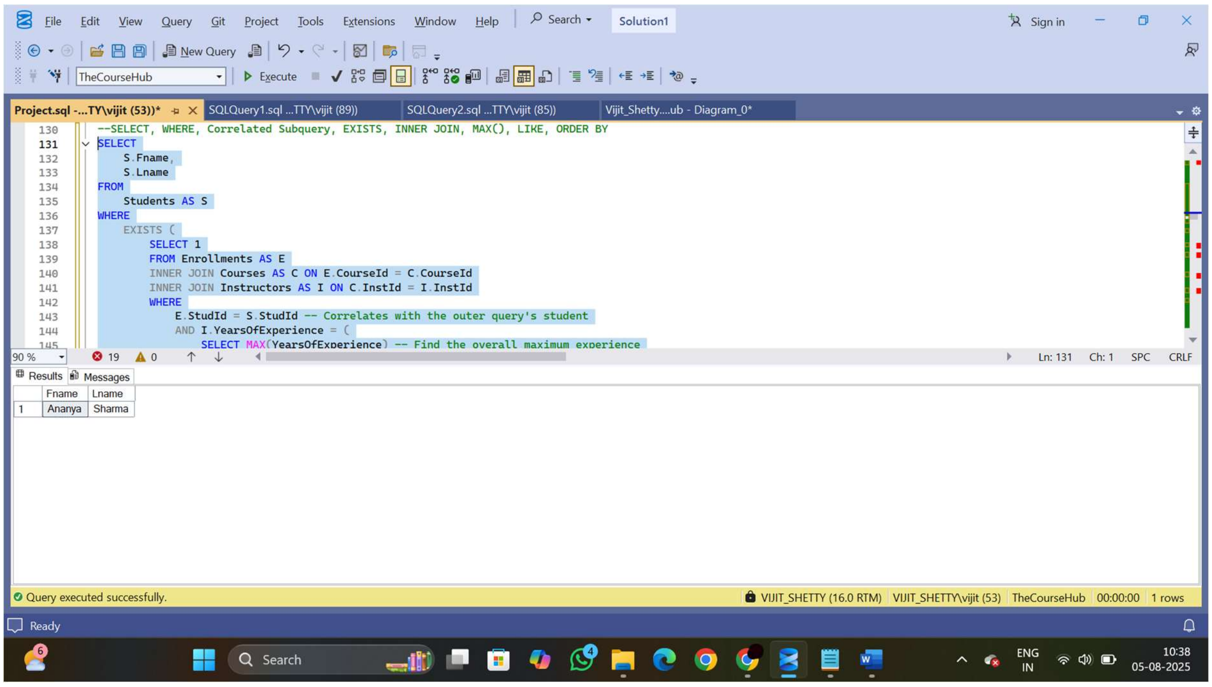
	fname	lname

Query executed successfully. VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 3 rows

10. Find the FirstName and LastName of students who are enrolled in any course taught by the instructor with the most years of experience from the student's own city.

Concept : SELECT, WHERE, Correlated Subquery, EXISTS, INNER JOIN, MAX(), ORDERBY

Snapshot :



```sql
--SELECT, WHERE, Correlated Subquery, EXISTS, INNER JOIN, MAX(), LIKE, ORDER BY
SELECT
    S.Fname,
    S.Lname
FROM
    Students AS S
WHERE
    EXISTS (
        SELECT 1
        FROM Enrollments AS E
        INNER JOIN Courses AS C ON E.CourseId = C.CourseId
        INNER JOIN Instructors AS I ON C.InstId = I.InstId
        WHERE
            E.StudId = S.StudId -- Correlates with the outer query's student
            AND I.YearsOfExperience = (
                SELECT MAX(YearsOfExperience) -- Find the overall maximum experience
```

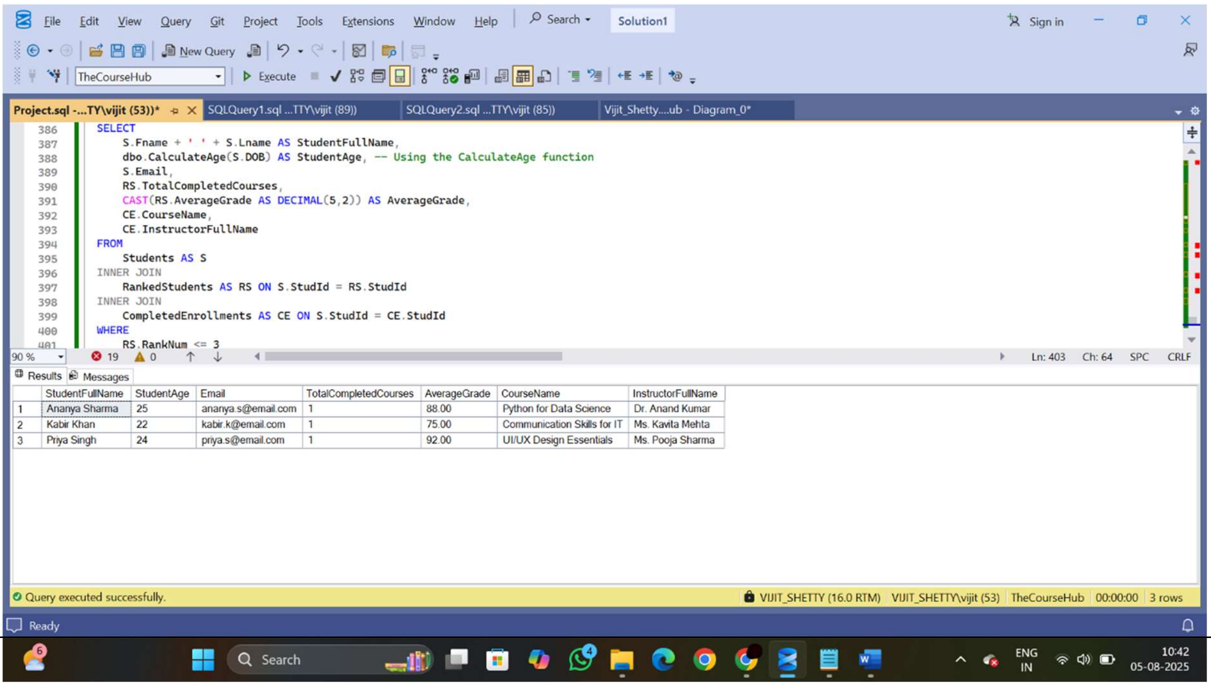
	Fname	Lname
1	Ananya	Sharma

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 1 rows

9. Identify the top 3 students by the total number of courses they are enrolled in. Then, for these top students, list their FullName, TotalCoursesEnrolled, and the Name of Each Course they are taking.

Concept : CTE (multiple WITH), SELECT, COUNT(), AVG(), GROUP BY, ROW_NUMBER() (Window Function), INNER JOIN (multiple), WHERE, ORDER BY, dbo.CalculateAge() (Function Call), CAST(), ISNULL(), String Concatenation.

Snapshot:



```sql
SELECT
    S.Fname + ' ' + S.Lname AS StudentFullName,
    dbo.CalculateAge(S.DOB) AS StudentAge, -- Using the CalculateAge function
    S.Email,
    RS.TotalCompletedCourses,
    CAST(RS.AverageGrade AS DECIMAL(5,2)) AS AverageGrade,
    CE.CourseName,
    CE.InstructorFullName
FROM
    Students AS S
INNER JOIN
    RankedStudents AS RS ON S.StudId = RS.StudId
INNER JOIN
    CompletedEnrollments AS CE ON S.StudId = CE.StudId
WHERE
    RS.RankNum <= 3
```

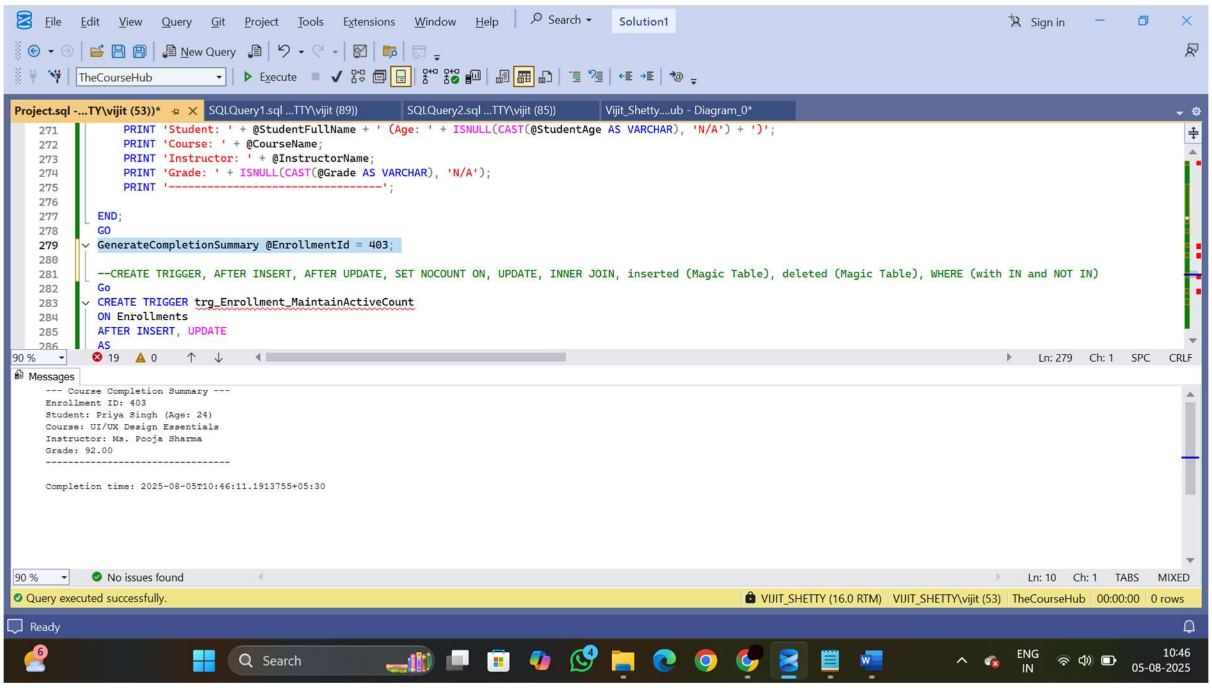
	StudentFullName	StudentAge	Email	TotalCompletedCourses	AverageGrade	CourseName	InstructorFullName
1	Ananya Sharma	25	ananya.s@email.com	1	88.00	Python for Data Science	Dr. Anand Kumar
2	Kabir Khan	22	kabir.k@email.com	1	75.00	Communication Skills for IT	Ms. Kavita Mehta
3	Priya Singh	24	priya.s@email.com	1	92.00	UI/UX Design Essentials	Ms. Pooja Sharma

Query executed successfully. VIJIT_SHETTY (16.0 RTM) VIJIT_SHETTY\vijit (53) TheCourseHub 00:00:00 3 rows

10. Get a course summary for each student using his/her id

Concept: CREATE PROCEDURE, Input Parameter, DECLARE Variables, SELECT INTO Variables, INNER JOIN (multiple), dbo.FunctionName() (Function Call), IF...ELSE, RAISERROR, RETURN, PRINT, CAST(), ISNULL()

Snapshot:



```sql
        PRINT 'Student: ' + @StudentFullName + ' (Age: ' + ISNULL(CAST(@StudentAge AS VARCHAR), 'N/A') + ')';
        PRINT 'Course: ' + @CourseName;
        PRINT 'Instructor: ' + @InstructorName;
        PRINT 'Grade: ' + ISNULL(CAST(@Grade AS VARCHAR), 'N/A');
        PRINT '--------------------------------';

    END;
    GO
    GenerateCompletionSummary @EnrollmentId = 403;

    --CREATE TRIGGER, AFTER INSERT, AFTER UPDATE, SET NOCOUNT ON, UPDATE, INNER JOIN, inserted (Magic Table), deleted (Magic Table), WHERE (with IN and NOT IN)
    Go
    CREATE TRIGGER trg_Enrollment_MaintainActiveCount
    ON Enrollments
    AFTER INSERT, UPDATE
    AS
```

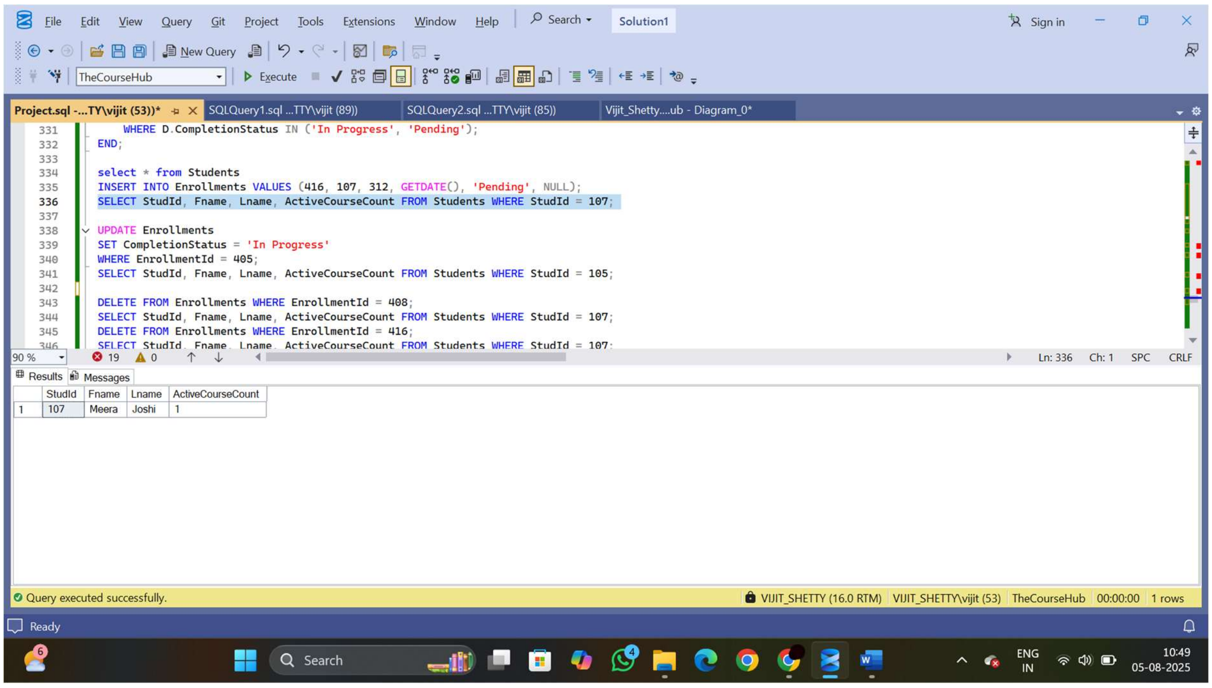
```
--- Course Completion Summary ---
Enrollment ID: 403
Student: Priya Singh (Age: 24)
Course: UI/UX Design Essentials
Instructor: Ms. Pooja Sharma
Grade: 92.00
--------------------------------

Completion time: 2025-08-05T10:46:11.1913755+05:30
```

Query executed successfully.

11. Tracking Student Engagement with "Active Course Count"

Concept: CREATE TRIGGER, AFTER INSERT, AFTER UPDATE, SET NOCOUNT ON, UPDATE, INNER JOIN, inserted (Magic Table), deleted (Magic Table), WHERE (with IN and NOT IN)

Snapshot :



```sql
    WHERE D.CompletionStatus IN ('In Progress', 'Pending');
END;

select * from Students
INSERT INTO Enrollments VALUES (416, 107, 312, GETDATE(), 'Pending', NULL);
SELECT StudId, Fname, Lname, ActiveCourseCount FROM Students WHERE StudId = 107;

UPDATE Enrollments
SET CompletionStatus = 'In Progress'
WHERE EnrollmentId = 405;
SELECT StudId, Fname, Lname, ActiveCourseCount FROM Students WHERE StudId = 105;

DELETE FROM Enrollments WHERE EnrollmentId = 408;
SELECT StudId, Fname, Lname, ActiveCourseCount FROM Students WHERE StudId = 107;
DELETE FROM Enrollments WHERE EnrollmentId = 416;
SELECT StudId, Fname, Lname, ActiveCourseCount FROM Students WHERE StudId = 107;
```

	StudId	Fname	Lname	ActiveCourseCount
1	107	Meera	Joshi	1

Query executed successfully.

VIJIT_SHETTY (16.0 RTM) | VIJIT_SHETTY\vijit (53) | TheCourseHub | 00:00:00 | 1 rows