Erasmus Student Exchange Tracker – Project Documentation-Student:ISXNLQ

# 1. Project Overview

The Erasmus Student Exchange Tracker is a relational database system designed to manage Erasmus student exchanges between universities. It tracks student information, university data, program participation, course offerings, and enrollment records.

# 2. ER Model Description

Entities:

* - Student
* - University
* - ExchangeProgram
* - Course
* - Enrollment

Relationships:

- A Student participates in one or more ExchangePrograms.

- An ExchangeProgram has a Home and Host University.

- A Course is offered by a Host University.

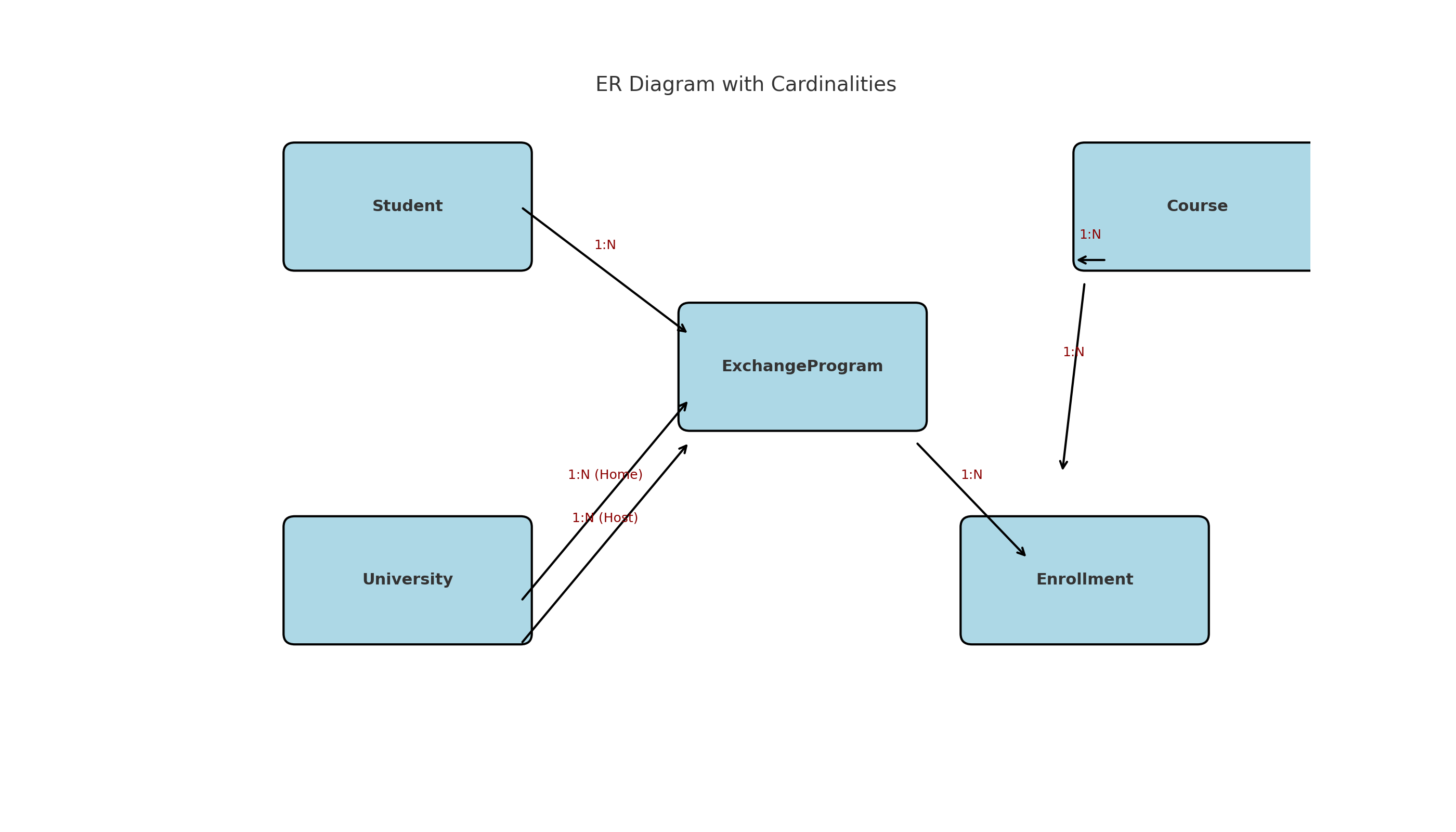
- Students are enrolled in Courses through Enrollments.

Below is a visual representation of the relationships between the entities in the Erasmus Tracker system.

## 2.1 Entity-Relationship Diagram

# 2.1 Entity-Relationship Diagram

Below is a visual representation of the relationships between the entities in the Erasmus Tracker system, including cardinalities.



University(UniversityID, Name, City, Country)

ExchangeProgram(ProgramID, StudentID, HomeUniversityID, HostUniversityID, StartDate, EndDate)

Course(CourseID, HostUniversityID, CourseName, Credits)

Enrollment(EnrollmentID, ProgramID, CourseID, Grade)

# 4. Stored Procedure

The procedure 'EnrollStudent' inserts a new enrollment record for a student into the Enrollment table based on the ProgramID and CourseID. It also commits the transaction.

CREATE OR REPLACE PROCEDURE EnrollStudent (  
 p\_ProgramID IN INT,  
 p\_CourseID IN INT,  
 p\_Grade IN VARCHAR2  
) AS  
 v\_NewEnrollmentID INT;  
BEGIN  
 SELECT NVL(MAX(EnrollmentID), 0) + 1 INTO v\_NewEnrollmentID FROM Enrollment;  
 INSERT INTO Enrollment (EnrollmentID, ProgramID, CourseID, Grade)  
 VALUES (v\_NewEnrollmentID, p\_ProgramID, p\_CourseID, p\_Grade);  
 COMMIT;  
END;  
/

# 5. Trigger (Not Executed)

Due to Oracle restrictions, triggers cannot be created under the SYS user. Below is the intended trigger that would log each enrollment into a separate log table.

CREATE TABLE EnrollmentLog (  
 LogID INT PRIMARY KEY,  
 EnrollmentID INT,  
 LogTime DATE  
);  
  
CREATE OR REPLACE TRIGGER trg\_LogEnrollment  
AFTER INSERT ON Enrollment  
FOR EACH ROW  
BEGIN  
 INSERT INTO EnrollmentLog  
 VALUES (  
 (SELECT NVL(MAX(LogID), 0) + 1 FROM EnrollmentLog),  
 :NEW.EnrollmentID,  
 SYSDATE  
 );  
END;  
/

# 6. Sample Query

SELECT s.FirstName, s.LastName, u.Name AS HostUniversity, c.CourseName, e.Grade  
FROM Student s  
JOIN ExchangeProgram ep ON s.StudentID = ep.StudentID  
JOIN University u ON ep.HostUniversityID = u.UniversityID  
JOIN Enrollment e ON ep.ProgramID = e.ProgramID  
JOIN Course c ON e.CourseID = c.CourseID;

# 7. Final Notes

This project demonstrates the creation and use of a relational database in Oracle with table creation, data insertion, querying, stored procedures, and trigger logic. The trigger component was included for documentation purposes, as it could not be executed under the SYS user.