ASSIGNMENT Task 4

-- TASK 4.1

-- Write an SQL query to calculate the average number of students enrolled in each course. Use

-- aggregate functions and subqueries to achieve this.

SELECT AVG(StudentCount) AS Avg\_Enrollment

FROM (

SELECT Course\_id, COUNT(Student\_id) AS StudentCount

FROM Enrollments

GROUP BY Course\_id -- the inner query is giving the count of students enrolled in each course and also grouping

) AS CourseEnrollments; -- now my outer query is using this data to calculate average

-- TASK 4.2

SELECT S.first\_name, S.last\_name, P.amount

FROM Payments AS P

INNER JOIN Students AS S

ON P.student\_id = S.student\_id

WHERE P.amount = (

SELECT MAX(amount)

FROM Payments -- fetches the maximum payment amount from the Payments table and also this is my inner query

);

-- TASK 4.1 Alternate

SELECT C.course\_id, c.course\_name , AVG(Enrollments) as Average

FROM Courses C

JOIN (SELECT course\_id, count(student\_id) as Enrollments

FROM Enrollments

GROUP BY course\_id)

e on C.course\_id = e.course\_id

GROUP BY C.course\_id, C.course\_name;

-- TASK 4.3

SELECT C.course\_id, C.course\_name, COUNT(E.student\_id) AS enrollment\_count

FROM Courses as C

LEFT JOIN

Enrollments as E ON C.course\_id = E.course\_id

GROUP BY C.course\_id,C.course\_name

HAVING COUNT(E.student\_id) = (SELECT MAX(enrollment\_count)

FROM (SELECT COUNT(student\_id) AS enrollment\_count

FROM enrollments

GROUP BY course\_id) AS subquery);

-- Task 4.4

SELECT T.First\_name, T.Last\_name,

(SELECT SUM(P.Amount)

FROM Payments P

JOIN Enrollments E ON P.Student\_id = E.Student\_id

WHERE E.Course\_id IN (

SELECT C.Course\_id

FROM Courses C

WHERE C.Teacher\_id = T.Teacher\_id

)

) AS Total\_Payments

FROM Teachers T;

-- TASK 4.5

SELECT S.First\_name, S.Last\_name

FROM Students S

WHERE (SELECT COUNT(E.Course\_id)

FROM Enrollments E

WHERE E.Student\_id = S.Student\_id

) = (SELECT COUNT(C.Course\_id)

FROM Courses C);

-- AS per my table there is no student who is enrolled in all the courses

-- TASK 4.6

SELECT T.First\_name, T.Last\_name

FROM Teachers T

WHERE T.Teacher\_id NOT IN (SELECT C.Teacher\_id FROM Courses C);

-- TASK 4.7

SELECT AVG(age) AS average\_age

FROM (

SELECT TIMESTAMPDIFF(YEAR, Date\_of\_Birth, CURDATE()) AS age

FROM Students

) AS subquery;

-- TASK 4.8

SELECT \*

FROM Courses

WHERE course\_id NOT IN (

SELECT Course\_id

FROM Enrollments

);

-- TASK 4.9

SELECT S.First\_name, S.Last\_name, C.Course\_name,

(SELECT SUM(P.Amount)

FROM Payments P

WHERE P.Student\_id = S.Student\_id

) AS Total\_Payments

FROM Students S

JOIN Enrollments E ON S.Student\_id = E.Student\_id

JOIN Courses C ON E.Course\_id = C.Course\_id;

-- TASK 4.10

SELECT S.First\_name, S.Last\_name

FROM Students S

WHERE (SELECT COUNT(P.Payment\_id)

FROM Payments P

WHERE P.Student\_id = S.Student\_id) > 1;

-- TASK 4.11

SELECT S.First\_name, S.Last\_name,

(SELECT SUM(P.Amount)

FROM Payments P

WHERE P.Student\_id = S.Student\_id

) AS Total\_Payments

FROM Students S;

-- TASK 4.12

SELECT C.Course\_name,

(SELECT COUNT(E.Student\_id)

FROM Enrollments E

WHERE E.Course\_id = C.Course\_id

) AS Enrolled\_Students

FROM Courses C;

-- TASK 4.13

SELECT S.First\_name, S.Last\_name,

(SELECT AVG(P.Amount)

FROM Payments P

WHERE P.Student\_id = S.Student\_id

) AS Average\_Payment

FROM Students S;