1. Retrieve a Student’s Overall Progress in a Course

SELECT

u.user\_id,

u.name,

c.title AS course\_title,

COUNT(a.assignment\_id) AS total\_assignments,

COUNT(p.assignment\_id) AS completed\_assignments,

ROUND(COUNT(p.assignment\_id) / COUNT(a.assignment\_id) \* 100, 2) AS completion\_percentage

FROM users u

JOIN progress p ON u.user\_id = p.user\_id

JOIN assignments a ON a.assignment\_id = p.assignment\_id

JOIN modules m ON m.module\_id = a.module\_id

JOIN courses c ON c.course\_id = m.course\_id

WHERE u.user\_id = 1

GROUP BY u.user\_id, c.course\_id;

2 List Assignments and Grades for a Student

SELECT

a.title AS assignment\_title,

a.type,

a.max\_score,

p.score,

p.status,

p.completed\_at

FROM progress p

JOIN assignments a ON a.assignment\_id = p.assignment\_id

WHERE p.user\_id = 5;

1. Identify Students At-Risk (Low Progress or Grades)

SELECT

u.user\_id,

u.name,

COUNT(p.assignment\_id) AS completed\_assignments,

AVG(p.score) AS average\_score

FROM users u

JOIN progress p ON u.user\_id = p.user\_id

GROUP BY u.user\_id

HAVING COUNT(p.assignment\_id) < 3 OR AVG(p.score) < 50;

1. Weekly Activity Tracking for a Student

SELECT

WEEK(p.completed\_at) AS week\_number,

COUNT(\*) AS completed\_assignments

FROM progress p

WHERE p.user\_id = 1342 AND p.completed\_at IS NOT NULL

GROUP BY WEEK(p.completed\_at)

ORDER BY week\_number;

1. Course Completion Status for All Students

SELECT

u.user\_id,

u.name,

c.course\_id,

c.title AS course\_title,

COUNT(DISTINCT a.assignment\_id) AS total\_assignments,

COUNT(DISTINCT p.assignment\_id) AS completed\_assignments,

CASE

WHEN COUNT(DISTINCT p.assignment\_id) = COUNT(DISTINCT a.assignment\_id) THEN 'Completed'

ELSE 'In Progress'

END AS course\_status

FROM users u

JOIN permissions pe ON pe.user\_id = u.user\_id

JOIN courses c ON c.course\_id = pe.course\_id

JOIN modules m ON m.course\_id = c.course\_id

JOIN assignments a ON a.module\_id = m.module\_id

LEFT JOIN progress p ON p.assignment\_id = a.assignment\_id AND p.user\_id = u.user\_id

WHERE u.role = 'learner'

GROUP BY u.user\_id, c.course\_id;

1. Students Who Have Not Submitted a Particular Assignment

SELECT u.user\_id, u.name

FROM users u

JOIN permissions pe ON pe.user\_id = u.user\_id

JOIN courses c ON c.course\_id = pe.course\_id

JOIN modules m ON m.course\_id = c.course\_id

JOIN assignments a ON a.module\_id = m.module\_id

WHERE a.assignment\_id = 6754

AND u.role = 'learner'

AND NOT EXISTS (

SELECT 1 FROM progress p

WHERE p.user\_id = u.user\_id AND p.assignment\_id = a.assignment\_id

);

1. List Pending Assignments for a Learner

SELECT

a.assignment\_id,

a.title,

a.due\_date

FROM assignments a

JOIN modules m ON a.module\_id = m.module\_id

JOIN courses c ON m.course\_id = c.course\_id

-- Only include learners

JOIN users u ON u.role = 'learner'

-- Check for pending or missing progress

WHERE u.user\_id = 125

AND a.assignment\_id NOT IN (

SELECT assignment\_id

FROM progress

WHERE user\_id = u.user\_id AND status = 'completed'

);

1. List All Assignments with Their Status for a Learner

SELECT

a.assignment\_id,

a.title,

p.status,

p.score,

a.due\_date

FROM progress p

JOIN assignments a ON p.assignment\_id = a.assignment\_id

JOIN modules m ON a.module\_id = m.module\_id

JOIN courses c ON m.course\_id = c.course\_id

JOIN users u ON u.user\_id = p.user\_id

WHERE u.user\_id = 1490 AND u.role = 'learner';