1 Student table

```
CREATE TABLE student(
   student_id INT AUTO_INCREMENT PRIMARY KEY,
   stud_name varchar(50) NOT null,
   country varchar(50) NOT null,
   registration_Date date
  );
```

2 Course table

```
CREATE table course(
   course_id int AUTO_INCREMENT PRIMARY KEY,
   title varchar(50) NOT null,
   subject varchar(50) NOT null,
   level varchar(50) NOT null);
```

3 Enrollment table

```
CREATE TABLE enrollment(
   enroll_Date date,
   stud_id INT NOT null,
   c_id INT NOT null,
   FOREIGN KEY(stud_id) REFERENCES student(student_id),
   FOREIGN KEY(c_id) REFERENCES course(course_id)
   );
```

4 Progress table

```
CREATE TABLE progress(
   stud_id INT NOT null,
   c_id INT NOT null,
```

```
FOREIGN KEY(stud_id) REFERENCES student(student_id),
FOREIGN KEY(c_id) REFERENCES course(course_id),
completed_percent decimal(5,2) NOT null,
last_accessed date);
```

Assessment Tasks:

1. Find the most popular course per subject (by enrollments).

```
SELECT course.subject, enrollment.enroll_Date,COUNT(enrollment.stud_id) AS most_popular
FROM enrollment
INNER JOIN course ON enrollment.c_id=course.course_id
GROUP BY enrollment.enroll_Date
ORDER BY enrollment.stud_id,course.subject DESC;
```

2. List students who completed more than 80% in at least 3 courses.

```
SELECT student.stud_name,COUNT(progress.c_id) AS more_than_3
FROM progress
INNER JOIN student ON progress.stud_id=student.student_id
INNER JOIN course ON progress.c_id=course.course_id
WHERE progress.completed_percent>80
GROUP BY student.stud_name
HAVING  more_than_3>=3;
```

3. Calculate average course completion by level (e.g., beginner, intermediate).

```
SELECT AVG(progress.completed_percent) AS avg_percentage, course.level
FROM progress
INNER JOIN course ON course.course_id=progress.c_id
GROUP BY course.level;
```

4. Identify students inactive for more than 60 days.

```
SELECT student.student_id, student.stud_name, student.country, progress.last_accessed
FROM student
LEFT JOIN progress ON progress.stud_id = student.student_id
WHERE progress.last_accessed < DATE_SUB(CURDATE(), INTERVAL 60 DAY)
   OR progress.last_accessed IS NULL;
```

5. Determine the subject with the highest average completion rate.

```
SELECT course.subject,AVG(progress.completed_percent)AS avg_completion_rate
FROM progress
INNER JOIN course ON progress.c_id=course.course_id
GROUP BY course.subject
ORDER BY avg_completion_rate DESC;
```