

	Запрос	Отчет
1. Выводить список студентов по определённому предмету	<pre> SELECT s.first_name, s.last_name FROM Students s JOIN Enrollments e ON s.student_id = e.student_id JOIN Courses c ON e.course_id = c.course_id WHERE c.course_name = 'Название_предмета'; SELECT s.first_name, s.last_name FROM Students s JOIN Enrollments e ON s.student_id = e.student_id JOIN Courses c ON e.course_id = c.course_id WHERE c.course_name = 'Название_предмета'; </pre>	first_name,last_name
2. Выводить список предметов, которые преподает конкретный преподаватель	<pre> SELECT c.course_name FROM Courses c JOIN Teachers t ON c.teacher_id = t.teacher_id WHERE t.first_name = 'Имя' AND t.last_name = 'Фамилия'; </pre>	course_name
3. Выводить средний балл студента по всем предметам	<pre> SELECT AVG(g.grade) AS average_grade FROM Grades g JOIN Enrollments e ON g.enrollment_id = e.enrollment_id JOIN Students s ON e.student_id = s.student_id WHERE s.first_name = 'Имя' AND s.last_name = 'Фамилия'; </pre>	average_grade NULL
4. Выводить рейтинг преподавателей по средней оценке студентов	<pre> SELECT t.first_name, t.last_name, AVG(g.grade) AS average_grade FROM Teachers t JOIN Courses c ON t.teacher_id = c.teacher_id JOIN Enrollments e ON c.course_id = e.course_id JOIN Grades g ON e.enrollment_id = g.enrollment_id </pre>	first_name,last_name,average_grade
5. Выводить список преподавателей, которые преподавали более 3 предметов за последний год	<pre> SELECT t.first_name, t.last_name FROM Teachers t JOIN Courses c ON t.teacher_id = c.teacher_id WHERE YEAR(c.hire_date) &gt;= YEAR(CURDATE()) - 1 GROUP BY t.teacher_id HAVING COUNT(c.course_id) &gt; 3; </pre>	first_name,last_name
6. Выводить список студентов, которые имеют средний балл выше 4 по математическим предметам, но ниже 3 по гуманитарным	<pre> SELECT s.first_name, s.last_name FROM Students s JOIN Enrollments e ON s.student_id = e.student_id JOIN Grades g ON e.enrollment_id = g.enrollment_id JOIN Courses c ON e.course_id = c.course_id GROUP BY s.student_id HAVING AVG(CASE WHEN c.course_name LIKE '%математика%' THEN g.grade END) &gt; 4 AND AVG(CASE WHEN c.course_name LIKE '%гуманитарный%' THEN g.grade END) &lt; 3; </pre>	first_name,last_name

7. Определить предметы, по которым больше всего двоек в текущем семестре	<pre> SELECT c.course_name, COUNT(g.grade) AS count_of_twos FROM Grades g JOIN Enrollments e ON g.enrollment_id = e.enrollment_id JOIN Courses c ON e.course_id = c.course_id WHERE g.grade = 2 AND e.enrollment_date &gt;= DATE_SUB(CURDATE(), INTERVAL 6 MONTH) GROUP BY c.course_name ORDER BY count_of_twos DESC; </pre>	course_name,count_of_twos
8. Выводить студентов, которые получили высший балл по всем своим экзаменам, и преподавателей, которые вели эти предметы	<pre> SELECT s.first_name, s.last_name, t.first_name AS teacher_first_name, t.last_name AS teacher_last_name FROM Students s JOIN Enrollments e ON s.student_id = e.student_id JOIN Grades g ON e.enrollment_id = g.enrollment_id JOIN Courses c ON e.course_id = c.course_id JOIN Teachers t ON c.teacher_id = t.teacher_id WHERE g.grade = 5; </pre>	first_name,last_name,teacher_f irst_name,teacher_last_name
9. Просматривать изменение среднего балла студента по годам обучения	<pre> SELECT YEAR(e.enrollment_date) AS year, AVG(g.grade) AS average_grade FROM Enrollments e JOIN Grades g ON e.enrollment_id = g.enrollment_id WHERE e.student_id = (SELECT student_id FROM Students WHERE first_name = 'Имя' AND last_name = 'Фамилия') GROUP BY YEAR(e.enrollment_date) ORDER BY year; </pre>	year,average_grade
10. Определить группы, в которых средний балл выше, чем в других, по аналогичным предметам	<pre> SELECT e.course_id, AVG(g.grade) AS average_grade FROM Enrollments e JOIN Grades g ON e.enrollment_id = g.enrollment_id GROUP BY e.course_id HAVING average_grade &gt; (SELECT AVG(g2.grade) FROM Enrollments e2 JOIN Grades g2 ON e2.enrollment_id = g2.enrollment_id WHERE e2.course_id = e.course_id); </pre>	course_id,average_grade
11. Вставка записи о новом студенте	<pre> INSERT INTO Students (first_name, last_name, middle_name, birth_date, contact_info, enrollment_date) VALUES ('Имя', 'Фамилия', 'Отчество', 'YYYY-MM- DD', 'Контактные данные', 'YYYY-MM-DD'); </pre>	

12. Обновление контактной информации преподавателя	<pre> UPDATE Teachers SET contact_info = 'Новая контактная информация' WHERE teacher_id = 'Идентификатор преподавателя' OR (first_name = 'Имя' AND last_name = 'Фамилия');</pre>	
13. Удаление записи о предмете	<pre> DELETE FROM Grades WHERE enrollment_id IN (   SELECT enrollment_id   FROM Enrollments   WHERE course_id = 'Идентификатор предмета' );  -- Удаление записей о зачислении на предмет DELETE FROM Enrollments WHERE course_id = 'Идентификатор предмета';  -- Удаление самого предмета DELETE FROM Courses WHERE course_id = 'Идентификатор предмета';</pre>	
14. Вставка новой записи об оценке		